Eine modellgetriebene Vorgehensweise in der Data Warehouse Entwicklung

White Paper der solvistas GmbH

Thomas Neuböck
Jürgen Raab
Andreas Weißenböck

Linz, März 2014

1 Einleitung

Dieses White Paper stellt einen modellgetriebenen Ansatz zur Einführung eines Vorgehensmodells bei der Entwicklung von Data Warehouse (DWH) und Business Intelligence (BI) Systemen vor. Er wurde im Rahmen eines Beratungs- und Konzeptionsprojektes für ein Business Intelligence Competence Center (BICC) im Bereich einer öffentlichen deutschen Krankenversicherung entwickelt. Vergleichbar mit einer modellgetriebenen Vorgehensweise[1] in der Softwareentwicklung wurde für den DWH- und BI-Bereich ein ähnliches modellorientiertes Vorgehen mit agilen Elementen eingeführt. Neben der allgemeinen Projektvorgehensweise (agiler Ansatz), der fachlichen Sicht und der generellen Architektursicht, steht vor allem auch die Modellierung (konzeptuelle Modellierung, logische und physische Datenmodellierung) im Fokus der Betrachtung.

Ausgehend von den Anforderungen an ein neues Vorgehensmodell (VGM) in Abschnitt 2 werden vier verschiedene Dimensionen eines BI-Projektes in Abschnitt 3 betrachtet. In der Managementsicht und Projektvorgehensweise (Unterabschnitt 3.1) wird ein agiles Vorgehen propagiert. Während die fachliche Sichtweise (Unterabschnitt 3.2) die fachlichen Inhalte in einem BI- und DWH-Projekt in den Vordergrund stellt, wird in der Architektursicht (Unterabschnitt 3.3) die Architektur eines BI- und DWH-Systems betrachtet. In Unterabschnitt 3.4 wird der Fokus auf die Modellierung als wesentlicher Bestandteil des hier vorgestellten VGMs gelegt. Die wesentlichen Punkte dieses dargestellten Ansatzes werden im Abschnitt 4 als Fazit zusammengefasst.

[1] siehe beispielsweise (Poole J., 2001) und (Rodriguez A., Fernández-Medina E., Piattini M., 2007)

2 Anforderungen an ein neues VGM

Hinsichtlich Anforderungen an ein neues VGM wurden bezüglich der bisher eingesetzten Vorgehensweise folgende Fragen gestellt werden:

- Wie zufrieden sind unsere Kunden?
- Setzen wir das um und verstehen wir, was die Kunden wollen?
- Wie bewältigen wir fachliche und technische Komplexität?
- Wie sieht es mit der Transparenz und Verfügbarkeit von fachlichem und technischem Wissen aus?
- Haben wir eine Architektur (und eine Umsetzung), die einfach gewartet und erweitert werden kann?
- Haben wir ausreichende Richtlinien und Standards zur Qualitätssicherung und werden diese auch gelebt?
- Wie gehen wir bei geänderten Anforderungen oder aufgetretenen Hindernissen um?
- Wie hoch sind die Eigenständigkeit, das Verantwortungsbewusstsein und der Motivationsgrad im Team?

Aus der Beantwortung dieser Fragen ergeben sich folgende Forderungen an ein neues Vorgehensmodell:

- Kundenorientiert Umsetzung von Fachthemen
- Verringerung der Kommunikations- und Verständnislücke zwischen Fachlichkeit und Technik
- Verbesserter Umgang mit der fachlichen und technischen Komplexität durch Anforderungszerlegung und kurze Umsetzungszyklen
- Breite Wissensstreuung in der Abteilung
- Einführung und Umsetzung einer leicht wartbaren und erweiterbaren Architektur
- Qualitätsgesicherte Lösungen
- Flexibilität bei sich ändernden Anforderungen oder auftretenden Hindernissen
- Motivationsförderung in den Projektteams

3 Dimensionen in BI-Projekten

Die Durchführung von BI-Projekten verlangt die gleichzeitige Fokussierung in vier Dimensionen (Abbildung 1):

- Managementsicht und Projektvorgehensweise
- Fachliche Sicht
- Architektursicht
- Modellsicht

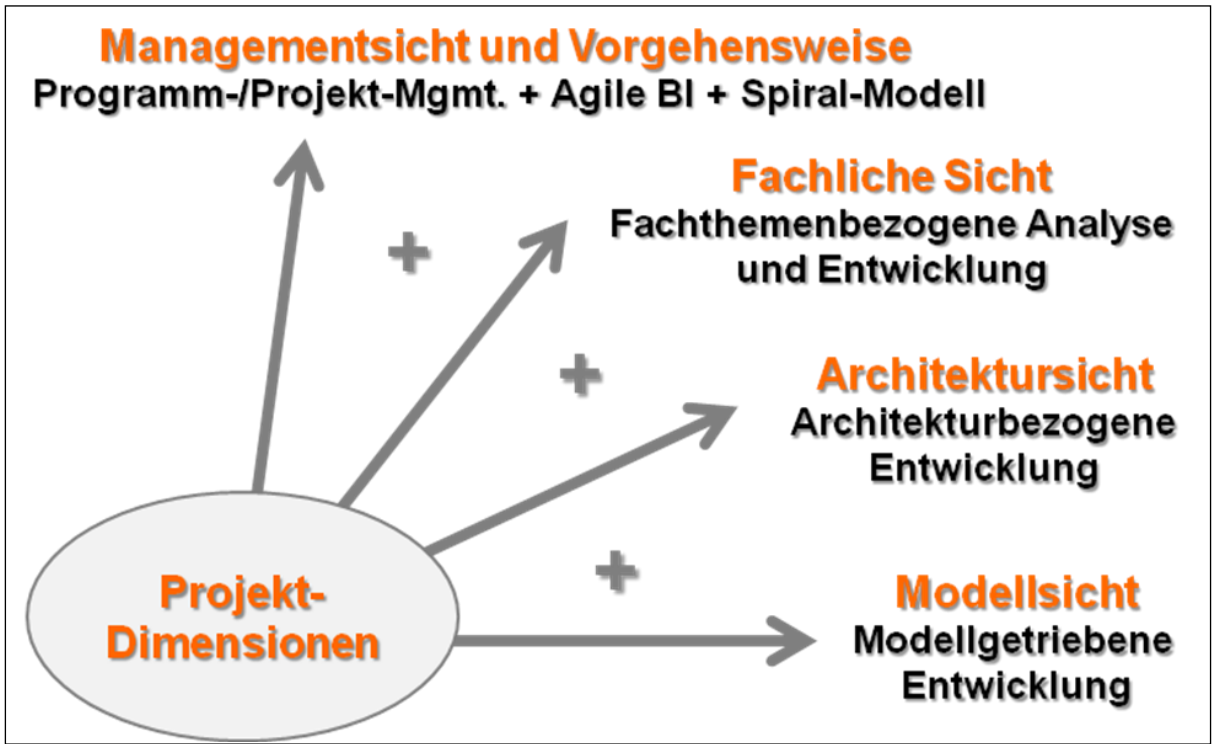

Abbildung 1

Die Managementsicht und Projektvorgehensweise verlangt ein übergeordnetes Programm- und Projektmanagement. Darin eingebettet werden Projekte und Programme in einer agilen Weise durchgeführt (agile BI). Komplexe Projekte oder Projektphasen mit hohem Erfolgsrisiko können im Sinne des Spiral-Modells[2] umgesetzt werden.

Die Tiefe von Fachthemen erfordert eine Zerlegungsstrategie, um Komplexität zu reduzieren. Verschiedene Themen verlangen auch unterschiedliche fachliche Sichtweisen. Mit dem VGM soll eine fachthemenbezogene DWH-Entwicklung erleichtert werden.

Im VGM ist die saubere Umsetzung und Pflege einer DWH-Architektur zu gewährleisten. DWH-Schichten sind sauber voneinander zu trennen. Zwischen den Schichten sind Abbildungsprozesse definiert.

Das vorgeschlagene VGM fordert eine modellgetriebene DWH-Entwicklung. Modelle erlauben die Einbringung verschiedener Sichtweisen und ergeben eine formale Dokumentation. Drei Modellebenen werden unterschieden:

- Computation Independent Model (CIM)
- Platform Independent Model (PIM)
- Platform Specific Model (PSM)

Das CIM beschreibt die Fachlichkeit, im PIM wird die logische plattformunabhängige und im PSM die plattformspezifische Umsetzung beschrieben.

Anmerkung:

Eine agile Vorgehensweise läuft in keiner ungeordneten und unkontrollierten Art und Weise ab. Richtlinien, Standards und Prozesse definieren den Rahmen, in dem eine agile Vorgehensweise eingebettet ist. Diese Rahmenbedingungen werden mit agilen Vorgehensweisen besser erfüllt - durch kurze Sprints, regelmäßige Reviews am Sprint-

[2] Zu Spiral-Modell nach Barry W. Boehm siehe zum Beispiel (Balzert H., 1998)

Ende, klare eindeutige Akzeptanzkriterien für User Stories und erhöhte Eigenverantwortung des Teams.

3.1 Managementsicht und Projektvorgehensweise

Wenngleich hier der Schwerpunkt auf die Möglichkeiten agiler Vorgehensweisen gelegt wird, soll in diesem Abschnitt das gesamte Umfeld begleitender Prozesse betrachtet werden.

3.1.1 Projektmanagement und begleitende Prozesse

Programm-Management, Projektmanagement und begleitende Prozesse, zu denen auch eine Kosten-/Nutzenbetrachtung gehört, bilden einen übergeordneten Rahmen für eine agile Vorgehensweisen. Begleitende Prozesse wie Anforderungsmanagement, Qualitätsmanagement oder Freigabeprozesse sind dabei wesentlich. Im Hinblick auf agile Vorgehensweisen übernimmt der Product Owner (unterstützt vom Scrum Master) eine wichtige Schnittstellenfunktion zum Projektmanagement und anderen begleitenden Prozessen.

An dieser Prozessschnittstelle werden Anforderungen in einem agilen Prozess (Sprint) umgesetzt. In einem Sprint arbeitet das agile Team eigenständig und selbstverantwortlich ohne Beeinflussung von außen. Ergebnisse werden vom agilen Prozess an die übergeordneten Prozesse (Programm-/Projektmanagement, begleitende Prozesse) übergeben.

Projektphasen können, falls dies sinnvoll erscheint, definiert werden. Beispielsweise wäre es denkbar auch weiterhin eine Gliederung in spezifische Phasen, wie beispielsweise „Entdecken“, „Entwerfen“, „Entwickeln“ und „Einführen“, anzuwenden – wenn auch sinnvollerweise in einer etwas abgewandelten Form. Die Entdeckenphase kann einer fachlichen Analyse entsprechen und wird in diesem Fall als separate Projektphase geführt. Tätigkeiten der Phasen „Entwerfen“ und „Entwickeln“ können in der Regel agil in Sprints abgewickelt werden. In diesem Fall werden dafür aber keine separaten Projektphasen definiert. Wird eine umfangreichere Gesamtlösung eingeführt, kann „Einführen“ als separate Phase betrachtet werden.

Anmerkung:

Die Wahrnehmung von Projektmanagementprozessen steht nicht im Widerspruch zu agilen Vorgehensweisen. Der Standard des Project Management Institute (PMI) definiert bspw. Projektmanagementprozessgruppen, erzwingt aber keine bestimmte Reihenfolge, wie diese Prozesse ablaufen.

3.1.2 Spiralmodell

Eine generelle Anforderung eines neuen VGM stellt die Komplexitätsbewältigung dar. Beim Spiralmodell handelt es sich um ein Metamodell, das eine inkrementelle prototyporientierte Entwicklung in mehreren Zyklen erlaubt. Auch im DWH- und BI-Bereich erscheint diese Vorgehensweisen in vielen Situationen angebracht.

Falls erforderlich wird in einem Zyklus eine prototyporientierte Umsetzung gestartet (z.B. als Proof of Concept). Nach einem abgeschlossenen Zyklus wird entschieden, wie mit der Umsetzung fortgefahren werden soll. Zu Beginn werden nur kleine Zyklen durchlaufen (geringe Kosten). Je mehr Zyklen abgeschlossen sind, umso höher ist auch die

Planungssicherheit und umso umfangreicher können die nächsten Zyklen vollzogen werden. Einzelne Zyklen können auch als separate Projektphasen definiert werden.

3.1.3 Agile Vorgehensweise

Agile Vorgehensweisen[3] haben in den letzten Jahren auch im BI- und DWH-Bereich Einzug gehalten. Die Möglichkeit über kleine Umsetzungszyklen schneller, effizienter und auch qualitativ hochwertiger die Projektziele zu erreichen, eröffnen einen besonderen Reiz gegenüber schwerfälligen rein phasenorientierten Prozessen. Ergebnisse werden schneller sichtbar und können mit dem Kunden diskutiert werden. Bei falschen Produktentwicklungen kann schneller gegengesteuert werden.

In den folgenden Abschnitten werden Grundprinzipien und Grundbegriffe allgemein erläutert. In späteren Kapiteln folgt eine unternehmensspezifische Betrachtung.

3.1.3.1 Manifest für Agile Softwareentwicklung

Ein Umdenken weg von starrem phasenorientierten bürokratischem Vorgehen hin zu agilen leichtgewichtigen Prozessen wurde von namhaften Personen eingeleitet. Im Februar 2001 wurde von 17 Unterzeichnern das folgende agile Manifest niedergeschrieben:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- ***Individuen und Interaktionen** mehr als Prozesse und Werkzeuge*
- ***Funktionierende Software** mehr als umfassende Dokumentation*
- ***Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung*
- ***Reagieren auf Veränderung** mehr als das Befolgen eines Plans*

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Unterzeichner: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Anmerkungen im Zusammenhang mit DWH- und BI-Systemen:

- ad „Individuen und Interaktionen mehr als Prozesse und Werkzeuge“: Auch zukünftig ist es wichtig Modellierungs- bzw. ETL-Werkzeuge einzusetzen. MitarbeiterInnen dienen nicht diesen Werkzeugen, sondern die Werkzeuge dienen den MitarbeiterInnen. Die Vermittlung von Zweck und Sinnhaftigkeit für den Einsatz dieser Werkzeuge und die Motivation zu deren Benutzung ist im besonderen Maße zu berücksichtigen.
- ad „Funktionierende Software mehr als umfassende Dokumentation“: Es stellt sich die Frage, ob dieses Prinzip nicht einer modellgetriebenen Architektur (MDA) bzw. modellgetriebenen Entwicklung widerspricht. Der Begriff „Software“ muss hier mit „DWH-System“ ersetzt werden. Modelle sind dabei als wichtige Bestandteile (auch im Sinne von Metadaten) eines „funktionierenden DWH-Systems“ zu betrachten und nicht als reine Dokumentation.

[3] siehe beispielsweise (Schwaber K., 2007), (Schwaber K., 2008) oder (Schwaber K., Beedle M., 2002).

- ad „Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung“: Ein DWH wächst getrieben von Fachthemen. Eine gute Zusammenarbeit mit den „Kunden“ ist dabei essentiell.
- ad „Reagieren auf Veränderung mehr als das Befolgen eines Plans“: Anforderungen können sich während der Entwicklung ändern. Vor der Frage „Tun wir das Richtige richtig?“ sollte zunächst immer Frage „Tun wir das Richtige?“ beantwortet werden. Wiederum gilt, dass die fachlichen Anforderungen im Vordergrund stehen. Organisatorische, budgetäre oder technische Rahmenbedingungen müssen natürlich dabei berücksichtigt werden. Häufige kleinere Iterationen in der Entwicklung mit abschließenden Evaluierungs- oder Review-Tätigkeiten erlauben es, auf Änderungen rasch und dennoch mit verhältnismäßig geringem Aufwand zu reagieren.

3.1.3.2 Prinzipien hinter dem Agilen Manifest

Die Unterzeichner des Agilen Manifests legten zwölf Prinzipien als Leitsätze fest:

„Wir folgen diesen Prinzipien:

- *Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.*
- *Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.*
- *Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.*
- *Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.*[4]
- *Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.*
- *Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteam zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.*
- *Funktionierende Software ist das wichtigste Fortschrittsmaß.*
- *Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.*
- *Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.*
- *Einfachheit - die Kunst, die Menge nicht getaner Arbeit zu maximieren*[5] *- ist essenziell.*
- *Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.*
- *In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.“*

[4] Die tägliche Zusammenarbeit zwischen Fachexperten und Entwicklern ist bereits ein Leitgedanke des BICCs.

[5] Dieses Prinzip ist kein „Freibrief zum Faulenzen“. Es geht darum, dass Arbeiten laufend hinsichtlich Nutzen hinterfragt werden. Nutzlose Tätigkeiten sind zu eliminieren. Einfachheit ist zu fördern – Komplexität ist nach Möglichkeit zu reduzieren. Die dadurch frei werdende Energie ist gewinnbringender einzusetzen.

Anmerkung:

Die Prinzipien einer agilen Vorgehensweise können nur dann erfolgreich eingesetzt werden, wenn alle Beteiligten (Management, Kunden, BI-Spezialisten, …) diese akzeptieren und davon auch überzeugt sind. Eine agile Vorgehensweise muss vom Management letztendlich getragen werden.

3.1.3.3 Iterationen und inkrementelle Entwicklung

In einer iterativen Vorgehensweise werden überschaubare Aufgaben definiert und in einem zeitlich ebenfalls überschaubaren Zyklus (Iteration) umgesetzt (ein bis vier Wochen). Am Ende einer Iteration wird das Ergebnis überprüft. Erfolgserlebnisse stellen sich sowohl für den Kunden als auch für das Entwicklungsteam schneller ein. Notwendige Anforderungsänderungen werden früher erkannt.

Ein DWH-System wird dabei inkrementell entwickelt. Es wächst ständig, nach jeder Iteration ist ein Teil dazugekommen. Mit der Umsetzung einer bestehenden Anforderung werden beim Kunden neue Bedürfnisse geweckt. Nachhaltige Systeme wie DWHs leben von diesem Zyklus – ein DWH ist in der Entwicklung nie abgeschlossen. Die DWH-Entwicklung orientiert sich in der Vorgehensweise am Spiral-Modell. Ein dafür vereinfachter Zyklus würde drei Aktivitäten beinhalten:

1. Zielfestlegung
2. Umsetzung
3. Evaluierung

Eine iterative und inkrementelle Vorgehensweise bedeutet nicht, dass ein langfristiger Gesamtplan obsolet ist. Dieser ist ohne Zweifel erforderlich und legt die Zielrichtung fest. Das Erreichen dieses Ziels wird in überschaubare Zyklen runtergebrochen, wobei zweckmäßige Korrekturen des Plans erlaubt sind.

Falls es sinnvoll erscheint, können auch bekannte Projektphasen (z.B. Entdecken – Entwerfen – Entwickeln – Einführen) mit iterativen Vorgehensweisen kombiniert werden. Z.B. könnte in der Phase „Entdecken“ eine gesamtheitliche Anforderungsanalyse betrieben werden, die Phasen „Entwerfen“ und „Entwickeln“ werden pro Iteration angewendet und „Einführen“ wieder als ungeteilte Phase ohne Iterationen. Oder es werden alle vier Phasen als Aktivitäten in einer Iteration untergebracht. Umgekehrt könnte wiederum jede dieser vier Phasen in einer Iteration abgehandelt werden.

3.1.3.4 Scrum-Elemente

Scrum ist ein agiles Vorgehensmodell, in dem iterativ und inkrementell agiert wird. Dabei werden die drei Prinzipien Transparenz, Überprüfung und Anpassung hoch gehalten.

Im Folgenden werden die grundlegenden Elemente von Scrum kurz erläutert.

3.1.3.4.1 Scrum-Rollen

Scrum unterscheidet sechs Rollen, von denen drei (Product Owner, Entwicklungsteam und Scrum Master) zum Scrum-Team zählen. Drei weitere Rollen (Kunde, Benutzer, Management) stellen externe Rollen dar, die außerhalb des Scrum-Teams wahrgenommen werden.

Anmerkung:

Im kundenspezifischen[6] Teil über Rollen wird der Begriff „Team“ verwendet, der hier in diesem allgemeinen Teil mit dem Begriff „Entwicklungsteam“ gleichzusetzen ist. Der Begriff „Entwicklungsteam“ wird im kundenspezifischen Teil bewusst nicht mehr weiterverwendet, um die Begrifflichkeit nicht allein auf die ETL-Entwicklung zu beschränken.

3.1.3.4.1.1 Product Owner

Der Product Owner ist für die strategische Produktentwicklung verantwortlich. Er ist für die Konzeption und Kommunikation einer klaren Produktvision, für die Festlegung und Priorisierung der jeweils zu entwickelnden Produkteigenschaften zuständig und entscheidet darüber, ob die vom Entwicklungsteam am Ende eines Sprints gelieferte Funktionalität akzeptabel ist. Der Product Owner ist kein Vertreter des Kunden. Er ist vielmehr „oberster Produktentwickler“, hält in dieser Funktion mit dem Kunden Kontakt und versucht, dessen Anforderungen in die Produktentwicklung einfließen zu lassen.

3.1.3.4.1.2 Entwicklungsteam

Das Entwicklungsteam ist für die Bereitstellung der vom Product Owner geforderten Produktfunktionalität und für die Einhaltung der vereinbarten Qualitätsstandards verantwortlich. Anforderungen (User Stories) werden in einem kurzen Zyklus von ein bis vier Wochen (Sprint) umgesetzt. Innerhalb eines Sprints arbeitet das Team autonom. Es organisiert sich selbst. Einflüsse von außen müssen unterbunden werden. Das Team umfasst idealerweise drei bis neun Personen und ist interdisziplinär zusammengesetzt.

3.1.3.4.1.3 Scrum Master

Der Scrum Master ist für das Gelingen von Scrum verantwortlich. Er arbeitet mit dem Team und Product Owner zusammen, ist aber selbst an der Sprint-Umsetzung nicht beteiligt. Der Scrum Master ist auch nicht der Chef des Teams. Er übernimmt vielmehr die Rolle eines Coach, Unterstützers und Beschützers war, ist für die Beseitigung von Hindernissen und für die Konfliktlösung verantwortlich.

3.1.3.4.1.4 Externe Rollen

Kunde, Benutzer und Management stellen externe Rollen dar. Der Product Owner stimmt Anforderungen mit dem Kunden ab. Der Benutzer verwendet schließlich das entwickelte Produkt. Er kann, muss aber nicht zugleich der Kunde sein. Das Management stellt die Rahmenbedingungen bereit, um Scrum anwenden zu können. Es muss hinter dem Scrum-Prozess stehen.

3.1.3.4.2 User Story

Eine User Story ist eine verbal formulierte Produktanforderung. Sie ist bewusst kurz gehalten und enthält folgende Informationen:

- Nutzer (als Rolle)
- Nutzen (als Begründung)
- Ergebnis (als Ziel)

Beispiele möglicher Textschablonen:

[6] bezogen auf das kundenspezifische Beratungsprojekt

„Als <Nutzer> möchte ich <Ergebnis>, um <Nutzen>."

oder:

„Als <Rolle> will ich <das Ziel>, so dass <Begründung>."

User Stories sind sehr feingranulare Anforderungen, die in Scrum im sog. Product Backlog gesammelt und von dort für einen Sprint ausgewählt und umgesetzt werden.

Eine User Story beschreibt Anforderungen aus Kundensicht (fachlicher Sicht), ist aber keine vollständige Dokumentation (oder Spezifikation), sie ist veränderlich und stellte eine Konversationsbasis dar.

Die User Story kann auf einer Story Card (im bewusst kleineren A5-Format) niedergeschrieben werden. Mit Akzeptanzkriterien (Definition of Done) wird sie konkretisiert (z.B. auf der Rückseite der Story Card). Wesentlich ist, dass die Konversation (zwischen Team, Product Owner und Kunde) als eine wichtige Komponente einer User Story betrachtet wird. Es steht nicht eine detaillierte komplexe schriftliche Spezifikation im Vordergrund, sondern die Konversation.

Die Güte einer User Story kann nach dem INVEST-Prinzip bewertet werden:

- **I**ndependent: ist möglichst unabhängig zu anderen User Stories
- **N**egotiable: ist verhandelbar, d.h. veränderbar
- **V**aluable: stellt einen Mehrwert für den Kunden dar
- **E**stimatable: ist ausreichend detailliert, um den Umsetzungsaufwand schätzen zu können
- **S**mall: ist kurz und bündig geschrieben
- **T**estable: die Erfüllung der Anforderungen kann getestet werden

User Stories sind zu priorisieren, müssen geschätzt und letztendlich geplant werden. Die Priorisierung kann bspw. in folgenden Kategorien erfolgen: Must have, Should have, Could have, Won't have this time. Die Schätzung kann in abstrakten Story Points (relative Schätzung) durchgeführt werden, wobei nur bestimmte Größenordnungen zur Aufwandsklassifizierung einer Story zulässig sind – z.B.: 1 Point, 3 Points, 5 Points, 8 Points, 13 Points, 20 Points. Die Abbildung zu konkreten Zeitdauern (Personentagen) erfolgt über Erfahrungswerte. Ist die Umsetzung einer User Story zu aufwändig bzw. zu komplex, muss sie weiter zerlegt werden („Schneiden einer User Story").

3.1.3.4.3 Scrum-Artefakte

3.1.3.4.3.1 Product Backlog

Der Product Backlog enthält alle umzusetzenden Anforderungen eines Produktes in Form von User Stories. Der Product Owner ist für den Product Backlog verantwortlich.

3.1.3.4.3.2 Sprint Backlog

Der Sprint Backlog enthält alle im aktuell laufenden Sprint umzusetzenden User Stories.

3.1.3.4.3.3 Burndown Charts

In Burndown Charts wird der Status des Product und des Sprint-Backlogs visualisiert. Man erhält einen Überblick wieviel bereits umgesetzt und wieviel noch offen ist.

3.1.3.4.3.4 Impediment Backlog

Das Impediment Backlog enthält eine Sammlung offener Probleme und Hindernisse. Es wird vom Scrum Master verwaltet. Dieser initiiert auch den Problemlösungsprozess.

3.1.3.4.3.5 Definition of Done

Die Definition of Done (DoD) ist eine Checkliste von Aktivitäten, die zur Implementierung einer User Story gehören und die Qualität beeinflussen. Sie ist schließlich auch die Grundlage für die Abnahme am Sprint-Ende. In der DoD sind zusätzlich zu den allgemein geltenden QS-Regeln die expliziten Qualitätsmerkmale dieser User Story aufzunehmen und entsprechend für eine Prüfung der QS zu dokumentieren und an einem vorgegebenen Ort abzulegen.

3.1.3.4.3.6 Definition of Ready

Die Definition of Ready (DoR) ist eine Checkliste, die bei der Erstellung der User Stories durch den Product Owner sowie bei deren Qualitätssicherung und spätestens bei der Übernahme von Stories vom Product ins Sprint Backlog Anwendung findet. Es können nur Stories in einem Sprint umgesetzt werden, die auch die DoR erfüllen. Die DoR verlangt, dass User Stories klar formuliert und umsetzbar (machbar) sind, und dass sie getestet werden können.

3.1.3.4.4 Meetings

Bei der folgenden Beschreibung der Meetings des Scrum-Prozesses ist zu erwähnen, dass der Scrum Master immer als Coach und Unterstützer (und tw. als Moderator) anwesend ist bzw. sein kann.

3.1.3.4.4.1 Sprint Planning Meeting 1

Das Sprint Planning Meeting 1 findet vor einem neuen Sprint statt. Der Product Owner wählt gemeinsam mit dem Entwicklungsteam die im neuen Sprint umzusetzenden User Stories aus. Diese User Stories werden vom Product Backlog in den Sprint Backlog übertragen. Als Dauer für das Sprint Planning Meeting 1 ist pro veranschlagte Sprint-Woche maximal eine Stunde vorzusehen.

3.1.3.4.4.2 Sprint Planning Meeting 2

Das Sprint Planning Meeting 2 findet anschließend an Sprint Planning Meeting 1 am selben Tag statt. Das Entwicklungsteam (und nur dieses) führt für den neuen Sprint eine Detailplanung durch und zerlegt die Umsetzung der User Stories in Tasks. Als Dauer für das Sprint Planning Meeting 2 ist pro veranschlagte Sprint-Woche maximal eine Stunde vorzusehen.

3.1.3.4.4.3 Daily Scrum Meeting

Das Daily Scrum Meeting sollte täglich für maximal 15 Minuten stattfinden. Es dient dem Entwicklungsteam (und nur diesem) als Möglichkeit des Informationsaustausches, damit der tägliche Status verfolgt werden kann und Problem möglichst rasch externalisiert werden können.

3.1.3.4.4.4 Sprint Review

Der Sprint Review steht am Ende eines Sprints. Als Dauer ist pro Sprint-Woche maximal eine Stunde vorgesehen. Im Sprint Review präsentiert das Entwicklungsteam dem Product

Owner die Sprint-Ergebnisse und dieser entscheidet, ob die User Stories des Sprints funktional und im Sinne der Definition of Done richtig und vollständig umgesetzt wurden.

3.1.3.4.4.5 Sprint Retrospektive

Unmittelbar nach dem Sprint Review folgt die Sprint Retrospektive, die nur dem Entwicklungsteam zugänglich ist. Es diskutiert den letzten abgeschlossenen Sprint, zeigt Probleme und Hindernisse auf und macht Verbesserungsvorschläge für die Zukunft. Probleme und Hindernissen werden im Impediment Backlog festgehalten. Als Dauer sind pro Sprint-Woche maximal 45 Minuten vorgesehen.

3.1.3.4.5 Sprint

Ein Sprint ist jener Prozess, in dem in einem kurzen Zyklus von ein bis vier Wochen aus dem Product Backlog ausgewählte User Stories umgesetzt werden. Er wird im Sprint Planning Meeting 1 und 2 geplant und am Ende im Sprint Review und in der Sprint Retrospektive bewertet. Innerhalb des Sprint-Prozesses arbeitet das Entwicklungsteam autonom und in Selbstverantwortung. Einflüsse von außen werden abgeschirmt, Product Owner, Kunde und Benutzer erhalten eine beratende Rolle.

3.1.3.4.6 Timeboxing

Timeboxing ist ein wichtiges Prinzip bei der Sprintdurchführung. Sowohl der Sprint selbst als auch die Meetings eines Sprints sind zeitlich streng begrenzt. Es wird damit entgegengewirkt, dass sich im Projekt Arbeiten „schleichend“ verzögern oder sich infolgedessen inhaltlich „schleichend“ reduzieren. Klare Akzeptanzkriterien bestimmen, ob eine User Story umgesetzt wurde oder nicht. Dementsprechend ist am „harten“ Endezeitpunkt eines Sprints ersichtlich, welche User Stories erfolgreich realisiert wurden und welche nicht. Nicht erfolgreich umgesetzte User Stories werden in einem späteren Sprint wieder aufgenommen. Der Projektfortschritt wird durch das Timeboxing daher regelmäßig neu hinterfragt. Dabei steht das Gesamtprojektziel immer im Vordergrund und nicht das Abschieben von Verantwortung oder die Suche nach Schuldzuweisungen. Ein Scrum-Team muss sich daher immer als geschlossene selbstverantwortliche organisatorische Einheit betrachten.

3.2 Fachliche Sichtweise

Eine BI bzw. ein DWH erfüllt keinen Selbstzweck sondern wird letztendlich von fachlichen Themen und Anforderungen getrieben. Aus den fachlichen Bereichen heraus entwickeln sich fachliche Themen. Zu jedem Thema werden die fachlichen Anforderungen analysiert. Auf Basis der fachlichen Anforderungen sind die technischen Anforderungen zu ermitteln.

Die fachliche Anforderungsanalyse wird sich primär auf die Anforderungen der Analyseschicht (im Sinne von Business Intelligence) konzentrieren. In erster Linie werden Geschäftsanwendungsfälle, fachliche Prozesse und fachliche Daten in dieser Ebene identifiziert. In einer weiteren Betrachtung kann der für die Analyseschicht erforderliche Informationsbeschaffungsprozess fachlich durchleuchtet werden.

Die Zerlegung von Anforderungen nach ihrer Fachlichkeit bietet eine weitere Möglichkeit der Komplexitätsreduktion. Bei der Umsetzung fällt die Konzentration auf einen fachlichen Teilaspekt. Insbesondere wird diese Dekompositionsstrategie durch kurze Iterationen einer agilen Vorgehensweise unterstützt. Durch einen flexiblen Aufbau der Architektur- und

Modellschichten (insbesondere die Modellierung nach Data Vault) ermöglicht eine leichte Erweiterbarkeit, sodass die Zentrierung auf kleinere fachliche Themen kein Problem darstellt.

3.3 Architektursicht

Die DWH-Architektur muss ein festes und stabiles technisches Fundament für die Umsetzung fachlicher Anforderungen bilden. Die Änderbarkeit und Ausbaufähigkeit sind hinsichtlich sich ändernder und neuer Anforderungen zu gewährleisten. DWH-Schichten müssen um fachliche Themen einfach erweitert werden können.

In der DWH-Architektur des Kunden sind folgende Schichten definiert:

- Präsentationsschicht
- Data Mart Schicht (DMS)
- Integrierte Datenschicht (IDS)
- Historisierte Datenschicht (HDS)
- Staging Area

3.4 Modellsicht

3.4.1 Modellgetriebene Entwicklung

Im Sinne einer Model Driven Architecture (MDA) wird das BI-/DWH-System in drei getrennten Modellschichten beschrieben:

- Computation Independent Model (CIM)
- Platform Independent Model (PIM)
- Platform Specific Model (PSM)

Insbesondere soll dadurch eine Separierung von fachlichen und technischen Betrachtungsweisen ermöglicht werden.

Motivation hinter der Einführung einer modellgetriebene Architektur beim Kunden:

- Eine modellgetriebene Architektur bietet mehrere Sichtweisen (fachliche Sichtweisen, technische Sichtweisen). Dem wird durch die drei Modellschichten (CIM, PIM, PSM) Rechnung getragen. Die Größe von DWH-Systemen erwirkt eine intensive Kommunikation verschiedener Akteure (Fachexperten, technische Experten, …). Verschiedene Akteure benötigen unterschiedliche Sichtweisen und Kommunikationsmitteln.
- Die Komplexität von DWH-Systemen ist groß und erhöht sich laufend mit der Zeit. Modelle auf verschiedenen Ebenen bieten Abstraktionen an, um diese Komplexität zu beherrschen.
- Fachliche Anforderungen werden zunächst in der obersten Modellschicht (CIM) modelliert und in die darunter liegenden transformiert (PIM, PSM). Damit wird eine fachgetriebene Entwicklung des DWHs gefördert und die Akzeptanz im Fachbereich garantiert.

Anmerkung:

Es ist zu beachten das Architektur- und Modellschichten begrifflich nicht vermischt werden, auch wenn beide Sichtweisen im Zusammenhang betrachtet werden können. So kann beispielsweise der Architekturschicht IDS aus Modellsicht fachlich (CIM), logisch (PIM) und plattformspezifisch (PSM) betrachtet werden.

Die Modellschichten können sowohl im Zusammenhang mit Fachthemen als auch im Zusammenhang mit den Architekturschichten betrachtet werden.

Modellperspektiven und alle Fachthemen:

Modellperspektiven und Modelle		Fachthemen
CIM	Anforderungsbeschreibung	
	Organisationsbeschreibung	
	Domänenmodell	
	Geschäftsanwendungsfälle	
	Geschäftsprozesse	
	Dimensionales Faktenmodell	
PIM	Systemanwendungsfälle	
	Benutzermodell	
	Logisches Datenmodell der DMS	
	Logisches Datenmodell der IDS	
	Logisches Datenmodell der HDS	
	Prozessbeschreibung der Systemanwendungsfälle	
	Beschreibung der Datenflüsse zwischen den logischen Architekturschichten	
	Plattformunabhängige Beschreibung von ETL-Prozessen	
PSM	Plattformspezifisches ETL-Prozessmodell	
	Physisches Datenbankschema der DMS	
	Physisches Datenbankschema der IDS	
	Physisches Datenbankschema der HDS	

Modellperspektiven und Architekturschichten:

Modellperspektiven und Architekturschichten		Architekturschicht		
		DMS	IDS	HDS
CIM	Anforderungsbeschreibung	X	X	X
	Organisationsbeschreibung	X	X	X
	Domänenmodell	X	X	X
	Geschäftsanwendungsfälle	X	X	X
	Geschäftsprozesse	X	X	X
	Dimensionales Faktenmodell	X	X	X
PIM	Systemanwendungsfälle	X	X	X
	Benutzermodell	X	X	X
	Logisches Datenmodell der DMS	X		
	Logisches Datenmodell der IDS		X	
	Logisches Datenmodell der HDS			X
	Prozessbeschreibung der Systemanwendungsfälle	X	X	X
	Beschreibung der Datenflüsse zwischen den logischen Architekturschichten	X	X	X
	Plattformunabhängige Beschreibung von ETL-Prozessen	X	X	X
PSM	Plattformspezifisches ETL-Prozessmodell	X		
	Physisches Datenbankschema der DMS	X		
	Physisches Datenbankschema der IDS		X	
	Physisches Datenbankschema der HDS			X

3.4.1.1 Abhängigkeiten und Transformationen

Hinsichtlich Abhängigkeiten und Transformationen innerhalb von und zwischen den Modellschichten sind folgende Überprüfungen durchzuführen:

- Es ist zu überprüfen, welche Abhängigkeiten und Transformationen zwischen den Modellen innerhalb einer Schicht (CIM, PIM, PSM) relevant sind.
- Es ist zu überprüfen, welche Abhängigkeiten und Transformationen von der CIM- in die PIM-Schicht und von der PIM- in die PSM-Schicht relevant sind.
- Es ist zu überprüfen, wie die Abhängigkeiten und Transformationen vom Modellierungsprozess und von den Werkzeugen unterstützt werden können.

3.4.1.2 Werkzeugunterstützung und Dokumentation

Die Modellschichten sind hinsichtlich Werkzeugunterstützung und Dokumentation zu überprüfen. Dies betrifft insbesondere Anforderungsmanagementsysteme, Modellierungswerkzeuge und Wissensmanagementsystem.

Ein Anforderungsmanagementsystem (z.B. Jira) wird als Project- und Issue-Tracking-Werkzeug verwendet. Anforderungen stellen in solchen Systemen häufig nur eine „fachliches Delta“ ab. Sie ermöglichen keine fachliche Gesamtsicht. Anforderungen werden häufig verbal erfasst. Ein erweiterter Einsatz von Anforderungsmanagementsystemen für die agile Vorgehensweise ist zu empfehlen (z.B. Jira Agile).

Konzeptuelle Modelle stellen Gesamtsichten auf die Fachlichkeit bereit. Modellierungswerkzeuge mit einer Verbindungsmöglichkeit zu fachlichen Anforderungen wäre dafür vorteilhaft. Organisationsbeschreibungen, Domänenmodelle, Geschäfts- bzw. BI-Anwendungsfälle, BI-Prozesse und Dimensionen-Fakten-Modelle erhöhen das fachliche Verständnis.

Logische Modelle beschreiben eine mögliche plattformunabhängige Umsetzung der Fachlichkeit. Darin enthaltene Systemanwendungsfälle erhöhen das Verständnis über die Funktionsweise des Systems. Von einem Modellierungswerkzeug wird daher die Darstellung von Anwendungsfalldiagrammen gefordert. Zusätzlich sind zu Beschreibung der Dynamik Aktivitäts-, Zustands-, Sequenz- und Kollaborationsdiagramme erwünscht.

Logische Datenmodelle der DMS, IDS und ev. HDS erhöhen das Verständnis über die Funktionsweise des Systems. Von einem Modellierungswerkzeug wird daher die Darstellung von logischen Datenmodellen gefordert.

Anforderungen sind sowohl für den Fachbereich als auch für das Umsetzungsteam relevant. Die Informationen zu den Anforderungen aus dem Anforderungsmanagementsystem und zu den Verbindungen zu den fachlichen Modellen aus dem Modellierungswerkzeug sind bereitzustellen – entweder über diese Werkzeuge selbst oder über separate Wissensmanagementsysteme (z.B. Confluence)

3.4.1.3 Computation Independent Model (CIM)

Im CIM werden die Fachlichkeit als Gesamtheit oder/und die fachlichen Anforderungen eines BI-/DWH-Systems dargestellt. Es werden konzeptuelle Modelle (fachliche Modelle, Geschäftsmodelle) erstellt. Folgende Beschreibungs- und Darstellungsformen erscheinen dafür als geeignet:

- Verbale Anforderungsbeschreibung und weitere verbale Beschreibungen (Geschäftsobjekte, Beziehungen, Regeln, Prozesse und Anwendungsfälle)
- Organisationsbeschreibung
- Konzeptuelle Modelle fachlicher Entitäten (Domänenmodell, Geschäftsobjekte), insbesondere Entity-Relationship-Modell (ER-Modell)
- Geschäftsanwendungsfälle
- Geschäftsprozesse, Analyseprozesse
- Konzeptuelle Modelle der Analyseschicht, insbesondere das Dimensional-Fact-Model (DFM)[7]

[7] siehe (Golfarelli M., Maio D., Rizzi S., 1998)

Anmerkung:

Die Bedeutung fachlicher Modelle darf nicht unterschätzt werden. Letztendlich stellen sie einen wichtigen Beitrag zur Verringerung der Kommunikations- und Verständnislücke dar.

3.4.1.3.1 (Verbale) Anforderungsbeschreibung

Anforderungen werden im ersten Schritt aufgenommen und zunächst rein verbal dokumentiert (Fachliche Anforderungssammlung). Die Anforderungen sind nach Fachthemen gruppiert. Werden Anforderungen eines neuen Fachthemas analysiert, ist je nach Umfang und Komplexität möglicherweise die Erstellung eines Fachkonzeptes erforderlich.

Anforderungen sind hinsichtlich ihres Status zu klassifizieren (mögliche mit der Qualitätssicherung abzustimmende Klassifizierung):

- Umgesetzte Anforderung
- Umzusetzende Anforderung
- Abgelehnte Anforderung
- Neue Anforderung, die noch in Begutachtung ist und deren Umsetzung weder beauftragt noch abgelehnt wurde.

3.4.1.4 Organisationsbeschreibung

Zu den fachlichen Anforderungen und zu den konzeptuellen Modellen sind die relevanten Informationen zur Organisation zu erfassen:

- Organigramm: Die Organisationsstruktur wird in einem Organigramm abgebildet.
- Akteure: Die mit dem System interagierenden Personen, Organisationseinheiten und umgebenden technischen Systemen werden als Akteure modelliert. Mögliche Hierarchien (IS-A-Hierarchien) zwischen den Akteuren werden dargestellt. Die Akteure werden in der Modellierung der Geschäftsanwendungsfälle weiterverwendet.

3.4.1.5 Domänenmodell

Im Domänenmodell werden relevante fachliche Objekte und deren Beziehung untereinander beschrieben. Als Darstellungsform werden Entity-Relationship-Diagramme (ER-Diagramm) verwendet.

3.4.1.6 Geschäftsanwendungsfälle/BI-Anwendungsfälle

Ein Anwendungsfall beschreibt wichtige Szenarien die von einem System mit Hilfe von Akteuren geleistet werden, um fachliche Ziele zu erreichen. Akteure werden dabei als Umfeld des Systems betrachtet. Bei Geschäftsanwendungsfällen wird der Inhalt im Rahmen der gesamten betroffenen Organisation betrachtet, unabhängig von einem konkreten technischen System. Im BI-Bereich sind daher dementsprechend BI-Anwendungsfälle zu dokumentieren.

Geschäftsanwendungsfälle und Akteure werden in einem Anwendungsfalldiagramm dargestellt. Die einzelnen Anwendungsfälle werden verbal beschrieben.

Im BI-Bereich sind insbesondere Anwendungsfälle der Analyse, Datenbereitstellung und Analyseergebnisweitergabe relevant.

Von einem Modellierungswerkzeug wird die Darstellung von Anwendungsfalldiagrammen gefordert. Zusätzlich sind zu Beschreibung der Dynamik Aktivitäts-, Zustands-, Sequenz- und Kollaborationsdiagramme erwünscht.

3.4.1.7 Geschäftsprozesse/BI-Prozesse

Geschäftsprozesse beschreiben, wie ein System auf Geschäftsebene intern operiert. Im BI-Bereich ist vor allem die Beschreibung von BI-Prozessen ein wichtiges Element zur fachlichen Kommunikation (zwischen Fachbereich und Umsetzungsteam). Kernprozesse wie Versorgungsmanagement oder entscheidungsunterstützende Controlling-Prozesse müssen daher ausreichende dokumentiert werden, um den Nutzen des BI- und DWH-Systems zu verstehen. Als Darstellungsform wird die Business Process Modeling Notation (BPMN) empfohlen. Alternative wäre auch die Darstellung in Aktivitätsdiagrammen denkbar.

3.4.1.8 Dimensionales Faktenmodell

Das Dimensional Fact Model (DFM) stellt das wichtigste konzeptuelle Datenmodell für die fachliche Betrachtung der Analyseschicht dar. Folgende Modellelemente sind zu erfassen.

- Fakten
- Dimensionsrollen
- Dimensionen
- Dimensionshierarchien (parallele und alternative)
- Hierarchieebenen
- Dimensionsattribute
- Nicht-Dimensionsattribute

Ein Fachthema kann häufig als Data Mart betrachtet werden. Dieser wird dann als DFM beschrieben.

3.4.2 Platform Independent Model (PIM)

In dieser Modellschicht werden fachliche Anforderungen und fachliche Modellelemente in logische aber noch plattformunabhängige Modelle abgebildet. Folgende Beschreibungsmittel werden dafür empfohlen:

- Systemanwendungsfälle
- Benutzermodell
- Plattformunabhängiger Überblick über die DWH-Architektur
- Logisches Datenmodell der DMS (relationale Abbildung des DFM; Data Marts)

- Logisches Datenmodell der IDS (mit Data-Vault-Modellierung)[8]
- Logisches Datenmodell der HDS
- Prozessbeschreibungen
    - Prozessbeschreibung der Systemanwendungsfälle
    - Beschreibung der Datenflüsse (Mappings) zwischen den logischen Archtekturschichten (HDS, IDS, DMS)
    - Plattformunabhängigen Beschreibung von ETL-Prozessen

3.4.2.1 Systemanwendungsfälle

Ein Anwendungsfall beschreibt wichtige Szenarien, die von einem System mit Hilfe von Akteuren geleistet werden, um fachliche Ziele zu erreichen. Akteure werden dabei als Umfeld des Systems betrachtet. Bei Systemanwendungsfällen steht das zu beschreibende System (BI-/DWH-System) im Vordergrund.

Geschäftsanwendungsfälle bzw. BI-Anwendungsfälle werden auf Systemanwendungsfälle abgebildet. Die Darstellung erfolgt in einem Anwendungsfalldiagramm. Die einzelnen Anwendungsfälle werden verbal beschrieben.

Systemanwendungsfälle sind sowohl für den Fachbereich als auch für das Umsetzungsteam relevant. Die Informationen zu Systemanwendungsfällen können aus dem Modellierungswerkzeug oder über ein separates Wissensmanagementsystem bereitgestellt werden.

3.4.2.2 Benutzermodell

Aus der Beschreibung der Organisationsstruktur und der Akteure wird ein Benutzermodell abgeleitet.

3.4.2.3 Plattformunabhängige Sicht der DWH-Architektur

Eine plattformunabhängige Sicht enthält die Gliederung (als Überblicksdiagramm) in die Präsentationsschicht, DMS, IDS, HDS und Staging Area.

3.4.2.4 Logisches Datenmodell der DMS

Data Marts werden in einem logischen Datenmodell dargestellt (relationale Abbildung des DFM).[9]

3.4.2.5 Logisches Datenmodell der IDS

Die IDS ist als Data Vault Modell dargestellt. Die notwendigen Informationen werden aus den logischen Datenmodellen der Analyseschicht (Data Marts) und aus der HDS herangezogen.

[8] siehe (Linstedt D., 2010)

[9] siehe beispielsweise (Kimball R, Ross M., 2013)

3.4.2.6 Logisches Datenmodell der HDS

Allgemein ist auch zu Bereitstellung eines logischen Datenmodells der HDS in Erwägung zu ziehen. Dieses Modell kann im Sinne eines System of Records einfach gehalten oder relational modelliert werden. Im letzteren ist die vollständige oder teilweise Übernahme des Datenmodells aus dem operativen System sinnvoll.

3.4.2.7 Prozessbeschreibung der Systemanwendungsfälle

Es werden Prozessbeschreibungen erstellt, welche die Bedienung und das Verhalten des Systems widerspiegeln. Im Wesentlichen soll die Dynamik der Systemanwendungsfälle modelliert werden. Zu berücksichtigen sind hier auch notwendige Aktivitäten zur Qualitätssicherung. Von einem Modellierungswerkzeug wird daher die Darstellung von Prozessbeschreibungen in BPMN (alternativ auch als Aktivitätsdiagramm) gefordert.

3.4.2.8 Beschreibung der Datenflüsse zwischen den logischen Architekturschichten

Es werden reine Datenflussdarstellungen zwischen Präsentationsschicht, DMS, IDS, HDS und Staging Area erzeugt. Von einem Modellierungswerkzeug wird daher die Darstellung von Datenflüssen gefordert.

3.4.2.9 Plattformunabhängige Beschreibung von ETL-Prozessen

Ausgehend von den Datenflüssen und den logischen Datenmodellen werden Mappings und ETL-Prozesse unabhängig von der konkreten Plattform beschrieben. Zu berücksichtigen sind hier auch Aktivitäten zur Sicherung der Datenqualität (Prüfregeln).

Mappings und ETL-Prozesse erhöhen das Verständnis über die Funktionsweise des Systems. Von einem Modellierungswerkzeug wird daher die Darstellung von Mappings und ETLProzessen z.B. in BPMN gefordert.

3.4.3 Platform Specific Model (PSM)

Plattformunabhängige Modelle werden in plattformspezifische übergeführt

- Plattformabhängiger Überblick über die DWH-Architektur
- Werkzeugspezifisches ETL-Prozessmodell
- Physisches Datenbankschema der DMS
- Physisches Datenbankschema der IDS
- Physisches Datenbankschema der HDS

3.4.3.1 Plattformabhängiger Überblick über die DWH-Architektur

Ausgehend von der plattformunabhängigen Beschreibung der DWH-Architektur wird ein plattformabhängiger Überblick gegeben.

3.4.3.2 Werkzeugspezifisches ETL-Prozessmodell

Plattformspezifische Modelle beschreiben die konkrete plattformbezogene Umsetzung der Fachlichkeit als ETL-Prozess. Eine entsprechende Unterstützung über plattformspezifische Werkzeuge wird gefordert.

3.4.3.3 Physisches Datenbankschema der DMS

Aus dem logischen Datenmodell der DMS wird das physische Datenbankschema generiert.

3.4.3.4 Physisches Datenbankschema der IDS

Aus dem logischen Datenmodell der IDS wird das physische Datenbankschema generiert.

3.4.3.5 Physisches Datenbankschema der HDS

Aus dem logischen Datenmodell HDS wird das physische Datenbankschema generiert.

3.4.4 Modellierungsleitfaden

Der hier beschriebene Leitfaden gibt vor, wie bei der Modellierung eines neuen Fachthemas vorgegangen werden soll. Dargestellt wird in diesem Abschnitt der Modellierungsprozess. Modellierungsrichtlinien sind in einem eigenen Dokument zu finden. Abbildung 2 zeigt den gesamten Prozess der Modellierung und ETL-Entwicklung auf grober Ebene.

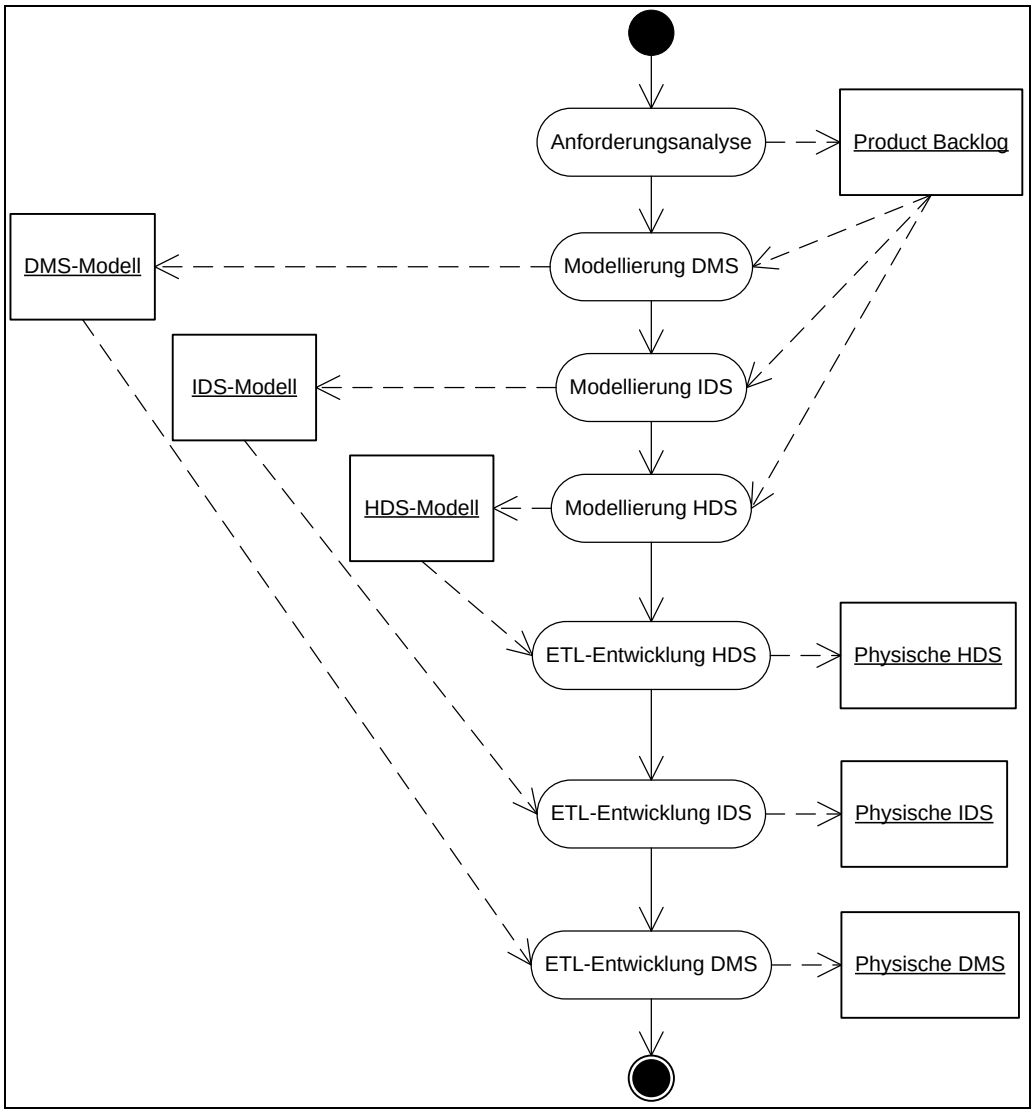

Abbildung 2

Aus der Anforderungsanalyse heraus werden Aufgaben in den Product Backlog übertragen. In Sprints werden Modellerweiterungen (Daten und Prozesse/Mappings; eine Modellierungsaktivität bzw. ein Modell der Architekturschicht beinhaltet in dieser groben Darstellung auch das Mapping in die nächste Schicht runter) durchgeführt. Es ist zu beachten, dass die Modelle Top-Down von der DMS zur IDS und abschließend zur HDS erweitert werden. Das wird damit gerechtfertigt, dass im BI-Bereich die Fachlichkeit und die Anforderungen sich zuerst in der DMS, anschließend in der IDS und abschließend in der HDS niederschlagen (Anm.: Die Top-Down-Betrachtung von der CIM zu PIM und abschließend zu PSM ist unabhängig davon unbestritten). Die Modelle dienen als Vorgabe für die ETL-Entwicklung (ebenfalls gleichzeitige Betrachtung von Daten und Prozessen/Mappings). Die ETL-Entwicklung selbst wird Bottum-Up durchgeführt: HDS, IDS und abschließend DMS. Selbstverständlich werden auch die ETL-Entwicklungsaktionen über Aufgaben aus dem Product Backlog initiiert – dies wird hier jedoch nicht dargestellt.

Abbildung 3 stellt den Modellierungsprozess in einer bereits weiter verfeinerten Form dar. Insbesondere wird hier operativ zwischen Modellanalyse und Modellerweiterung unterschieden.

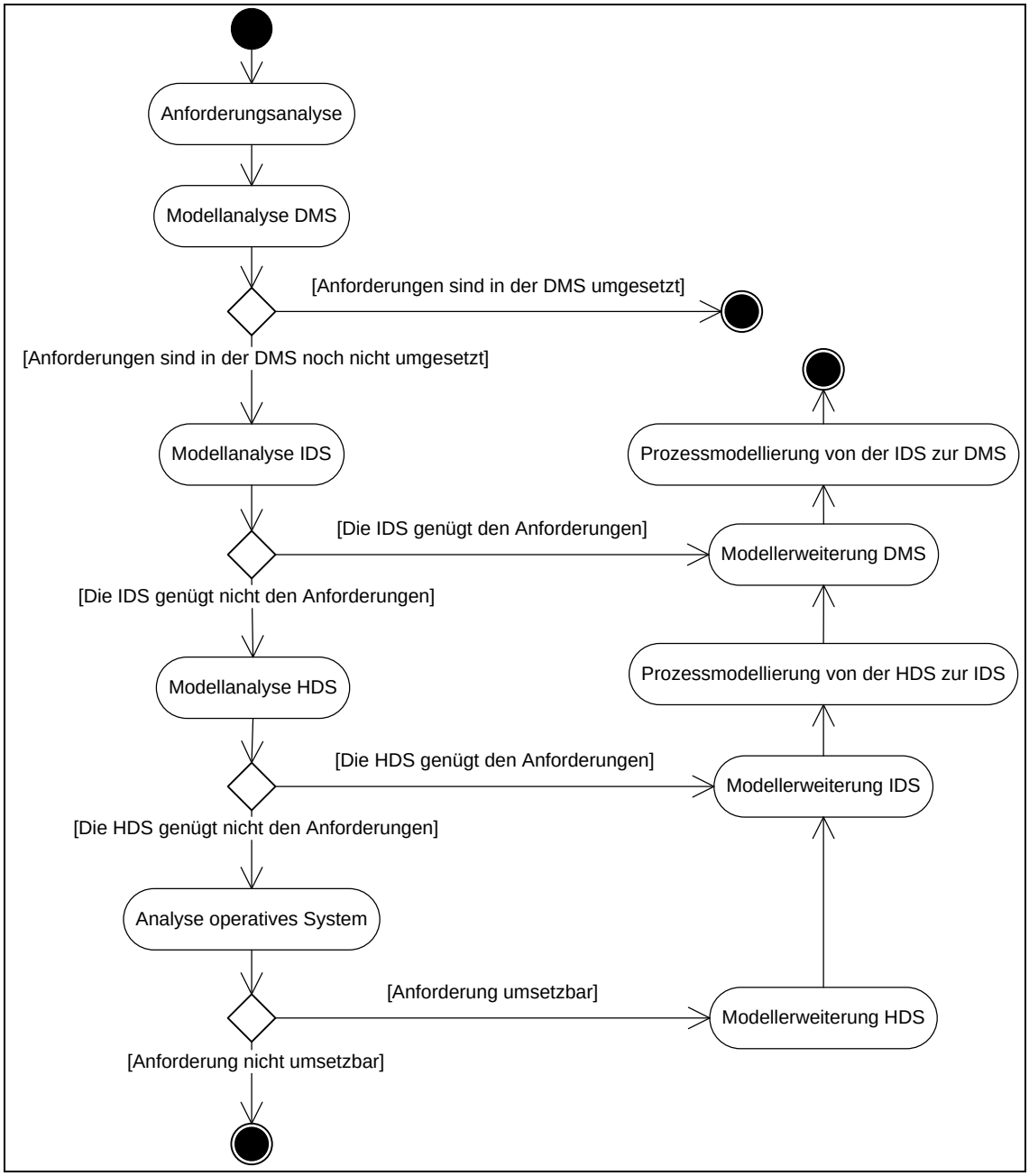
Anforderungsanalyse
Modellanalyse DMS
[Anforderungen sind in der DMS umgesetzt]
[Anforderungen sind in der DMS noch nicht umgesetzt]
Modellanalyse IDS
[Die IDS genügt den Anforderungen]
[Die IDS genügt nicht den Anforderungen]
Modellanalyse HDS
[Die HDS genügt den Anforderungen]
[Die HDS genügt nicht den Anforderungen]
Analyse operatives System
[Anforderung umsetzbar]
[Anforderung nicht umsetzbar]
Modellerweiterung HDS
Modellerweiterung IDS
Modellerweiterung DMS
Prozessmodellierung von der HDS zur IDS
Prozessmodellierung von der IDS zur DMS

Abbildung 3

3.4.4.1 Anforderungsanalyse

Als erster Schritt im Modellierungsprozess steht die Anforderungsanalyse eines neuen Fachthemas oder Teile eines neuen Fachthemas. Diese dient dann im Weiteren als Basis für die dazu erforderlichen Analysen der Präsentationsschicht und der DWH-Schichten (DMS, IDS und HDS). Es sind z.B. Kennzahlen, Berichte, Auswertungspfade (Auswertungshierarchien) und Dimensionen zu identifizieren und zu beschreiben.

Besonderheiten im kundenspezifischen Projekt:

- Ziel des Projektes ist es, die bestehenden Auswertungsmöglichkeiten im neuen Data Warehouse abzubilden. In diesem Sinne werden keine neuen Fachthemen oder fachlichen Anforderungen behandelt. Die bestehenden Data Marts im bestehenden (alten) Data Warehouse sind zu analysieren und ins neue Data Warehouse zu übernehmen.
- Als Ausgangsbasis dienen das bestehende BICC-Wiki und andere Dokumente. Es ist jedoch zu beachten, dass diese Dokumentation bei weitem nicht vollständig und teilweise auch veraltet ist.
- Die Analyse hat (zusätzlich) über die implementierten ETL-Jobs (im SAS Data Integration Studio) zu erfolgen, obwohl dies im Sinne einer Anforderungsanalyse eine nicht optimale Grundlage bietet.

3.4.4.2 Analyse und Erweiterung der BI-Anwendungsfälle

Abbildung 4 zeigt ein generisches und Abbildung 5 ein beispielhaftes Diagramm von BI-Anwendungsfällen.

Generisches Diagramm:

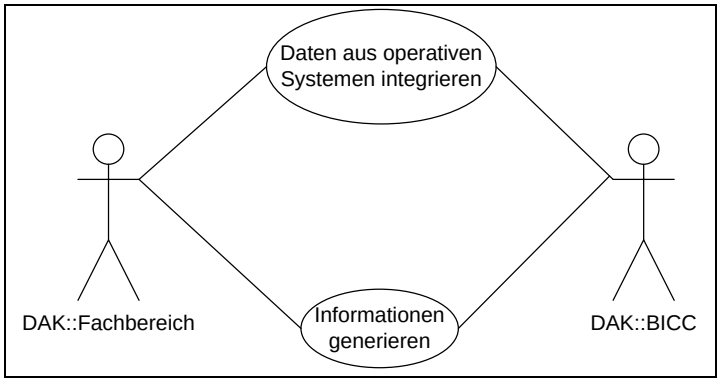

Abbildung 4

Beispieldiagramm:

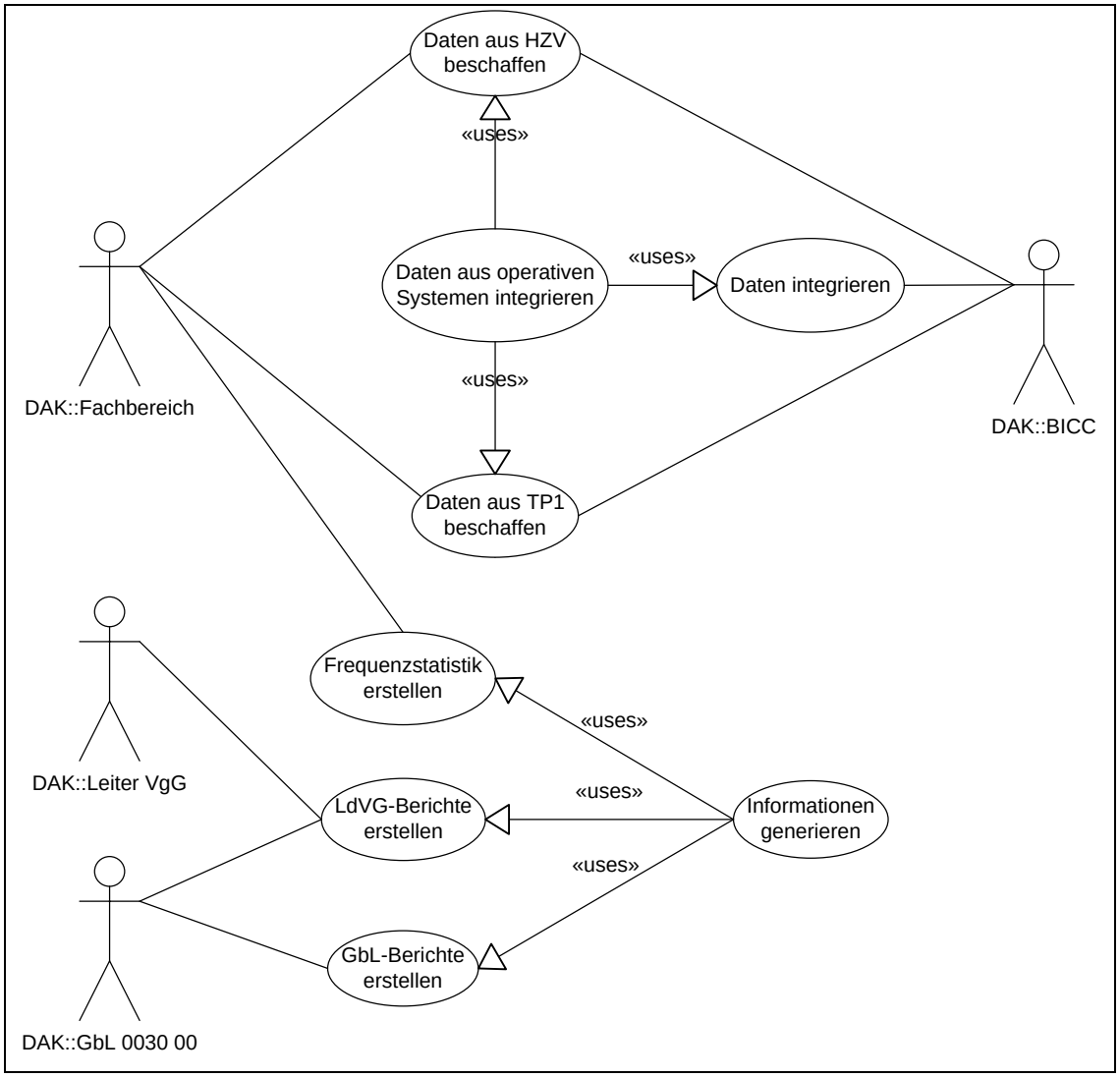
Daten aus HZV beschaffen
«uses»
Daten aus operativen Systemen integrieren
«uses»
Daten integrieren
«uses»
Daten aus TP1 beschaffen
DAK::Fachbereich
DAK::BICC
Frequenzstatistik erstellen
«uses»
DAK::Leiter VgG
LdVG-Berichte erstellen
«uses»
Informationen generieren
«uses»
GbL-Berichte erstellen
DAK::GbL 0030 00

Abbildung 5

3.4.4.3 Analyse und Erweiterung der fachlichen BI-Prozesse

Abbildung 6 zeigt ein generisches und Abbildung 7 ein beispielhaftes Diagramm von BI-Anwendungsfällen.

Generisches Diagramm:

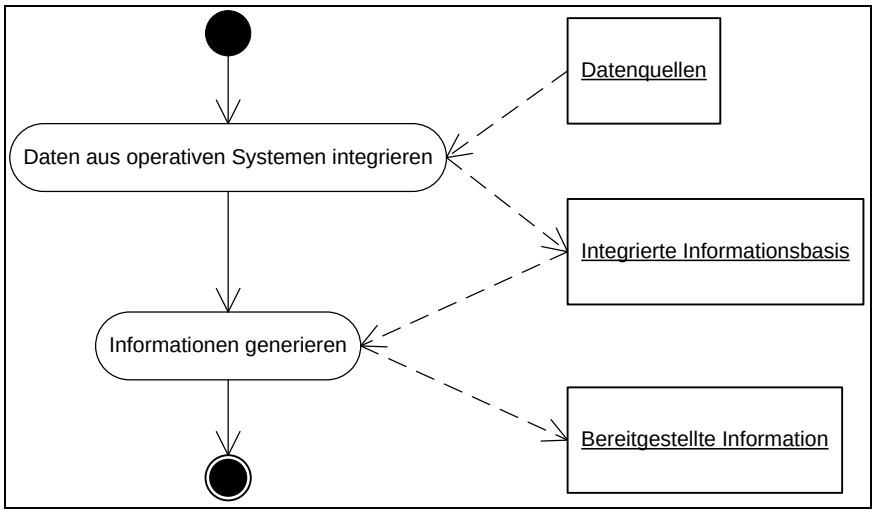

Abbildung 6

Beispieldiagramm:

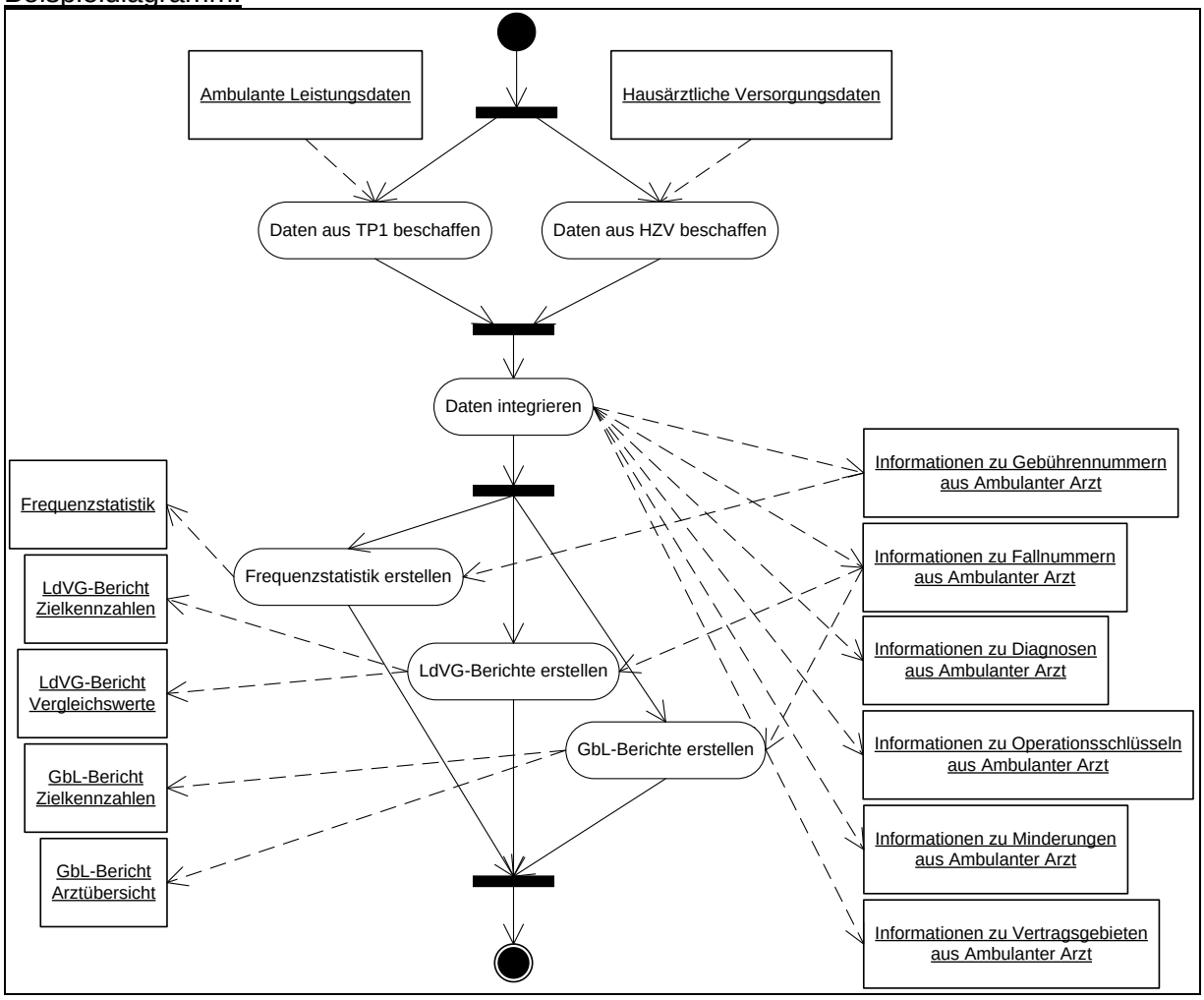

Abbildung 7

3.4.4.4 Analyse und Erweiterung des Domänenmodells

Das Domänenmodell ist dahingehend zu erweitern, dass neue Geschäftsobjekt und Beziehungen dazukommen. Eventuell müssen bestehende Beziehungen anpasst werden.

3.4.4.5 Analyse und Erweiterung des DFM-Modells

Das Domänenmodell ist dahingehend zu erweitern, dass neue Fakten, Dimensionen, Hierarchien oder Hierarchieebenen dazukommen. Eventuell müssen bestehende Elemente anpasst werden.

3.4.4.6 Analyse des DMS-Modells

Im weiteren Schritt des Modellierungsprozesses wird überprüft, ob die Anforderungen mit der bestehenden DMS bereits abgedeckt werden. Ist dies der Fall, sind keine weiteren Modellierungsschritte mehr erforderlich. Ansonsten wird zum Schritt „Erweiterung des DMS-Modells“ verzweigt.

Hinweis: Es wird nur die DMS modelliert. Die Modellierung einer Zugriffsschicht (Berichte usw.) erfolgt nicht.

Im Konkreten ist zu überprüfen, ob in der DMS alle erforderlichen

- Basiskennzahlen bzw. Fakten,
- Dimensionen und
- Dimensionshierarchien

zur Erfüllung der Anforderungen vorhanden sind.

Besonderheiten im kundenspezifischen Projekt:

- Die Informationen über Fakten und Dimensionen sind aus den Beschreibungen der Data Marts des bestehenden (alten) Data Warehouses zu entnehmen (sofern vorhanden).
- Die zugrunde liegende Dokumentation ist zu einem größeren Teil unvollständig und veraltet. Letztendlich sind die bestehenden Tabellen und ETL-Jobs im SAS Data Integration Studio zu analysieren.

3.4.4.7 Analyse des IDS-Modells

Nach der Analyse des DMS-Modells ist zu überprüfen, ob die Anforderungen in der IDS bereits umsetzbar sind. Ist dies der Fall, kann mit der Erweiterung des DMS-Modells fortgesetzt werden. Anderenfalls ist im nächsten Schritt die Analyse der HDS erforderlich.

Konkret ist zu überprüfen, ob alle für die Anforderungen notwendigen

- Dimensionen und
- Basiskennzahlen

der DMS aus

- Hubs,
- Satelliten,
- Links und
- ggf. Referenztabellen

der IDS generiert werden können.

Außerdem ist zu überprüfen, ob die Abbildungsprozesse von der IDS zur DMS zur Erfüllung der Anforderungen modelliert sind. Fehlen diese Abbildungsprozesse, ist zu analysieren, wie die erforderlichen Daten der DMS aus den Daten der IDS gewonnen werden können.

3.4.4.8 Analyse des HDS-Modells

Bei der Analyse der HDS sind alle Objekte zu identifizieren, die in der HDS noch fehlen, um die Anforderungen umzusetzen. Gibt es in der HDS keine fehlenden Objekte, kann direkt mit der Erweiterung des IDS-Modelles begonnen werden, ansonsten sind die operativen Systeme zu analysieren.

Im Zuge der Analyse des HDS-Modells ist zu analysieren, wie die Abbildungsprozesse von der HDS in die IDS auszusehen haben.

Besonderheiten im kundenspezifischen Projekt:

Um die Abbildungsprozesse von der HDS in die IDS zu finden, ist zu analysieren, wie im bestehenden (alten) Data Warehouse Fakten- und Dimensionstabellen aus den Tabellen der bestehenden (alten) HDS aufgebaut werden. Dazu sind die ETL-Jobs im SAS Data Integration Studio zu untersuchen. Es wird empfohlen, die daraus entstehenden Ergebnisse in einer Excel-Datei zusammenzufassen – mit folgenden Inhalten:

- Attribute der Zieltabelle (Dimensions- oder Faktentabelle)
- Kurze verbale Beschreibung der Attribute der Zieltabelle
- Mapping, aus welchen Attributen der Quelltabellen (HDS) das Attribute der Zieltabelle berechnet werden (einschließlich Beschreibung, wie die Berechnung aussieht)
- Join-Bedingungen und eventuell weitere zusätzliche Bedingungen
- Wird die Abbildungsbeschreibung zu komplex, wird empfohlen nicht eine einzige Mapping-Tabelle zu beschreiben, sondern zusätzliche Zwischentabellen zu führen und zu dokumentieren.

Diese Beschreibung in Excel-Form dient als Grundlage für die Erstellung des Prozessmodells von der HDS in die IDS (siehe 3.4.4.12). Der Modellierer muss aus dieser Dokumentation alle Informationen ablesen zu können, um ein umfassendes Prozessmodell generieren zu können, mit dem ein Entwickler die ETL-Jobs vollständig (ohne Zuhilfenahme der alten ETL-Jobs im SAS Data Integration Studio) ableiten kann.

3.4.4.9 Analyse des operativen Systems

Die fehlenden Objekte in der HDS müssen in den operativen Systemen eruiert werden. Sind nicht alle Objekte in der HDS konstruierbar, können die Anforderungen (oder Teile der Anforderungen) nicht umgesetzt werden. Im anderen Fall kann mit der Erweiterung des HDS-Modells fortgesetzt werden.

Besonderheiten im kundenspezifischen Projekt:

Um abschätzen zu können, welche Quelltabellen für die zukünftige HDS relevant sind, wird eine Zuordnung der Quelltabellen zum Component Business Model (CBM) erstellt.

3.4.4.10 Erweiterung des HDS-Modells

Fehlende Objekte der HDS, die zur Erfüllung der Anforderungen notwendig sind, müssen in der HDS ergänzt werden. Die in diesem Zusammenhang in den operativen Systemen identifizierten Objekte werden ins HDS-Modell übertragen.

Die Erweiterung des HDS-Modells erfolgt auf Basis der Analysen des operativen Systems (siehe 3.4.4.9) und der gegebenen HDS (3.4.4.8).

Schritte im Modellierungswerkzeug:

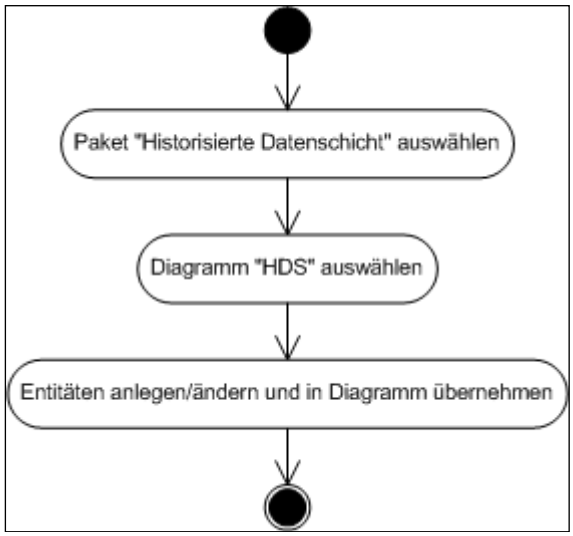

Abbildung 8

Im Modellexplorer des Innovators wird im Paket „Konzeptionelles Modell – Diagramme – Historisierte Datenschicht“ das Entity-Relationship-Diagramm „HDS“ ausgewählt. Darin sind die fehlenden Entitätstypen der HDS zu ergänzen oder bestehende zu ändern. Änderungen in einem Entitätstyp müssen nicht unbedingt über das Diagramm, sondern können auch direkt über das Modellelement erfolgen.

3.4.4.11 Erweiterung des IDS-Modells

Aufbauend auf HDS-Modell werden das IDS-Modell ergänzt, um die neuen Anforderungen umsetzen zu können.

Die Erweiterung des HDS-Modells erfolgt auf Basis der Analysen der HDS (siehe) und der gegebenen IDS (3.4.4.8).

Schritte im Modellierungswerkzeug:

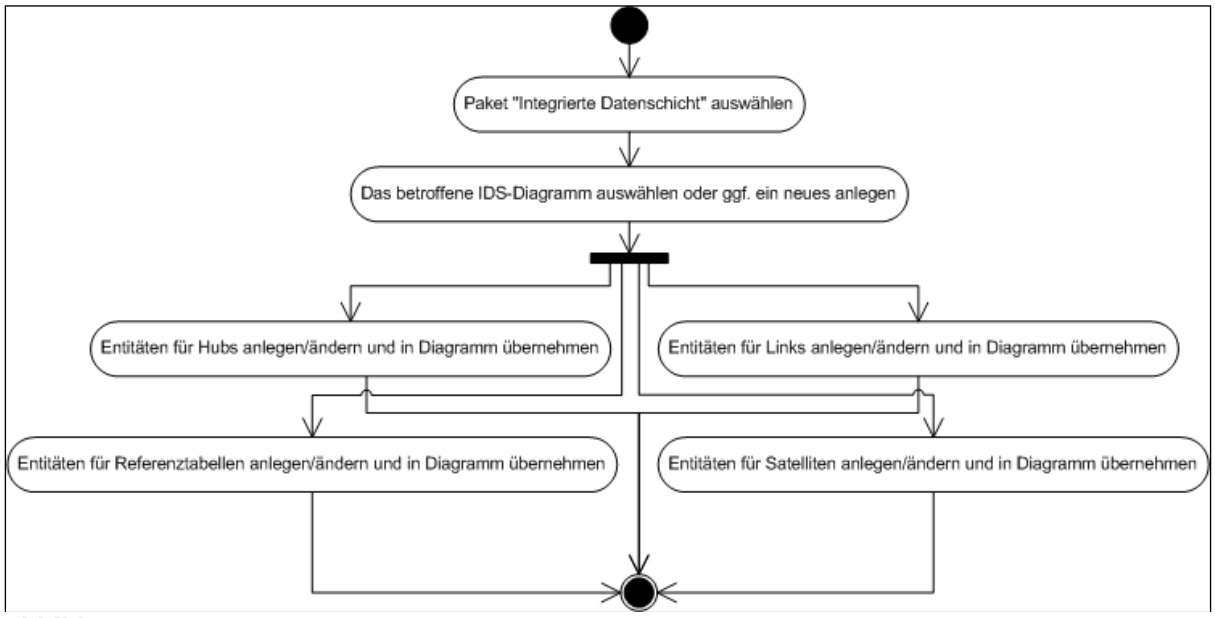

Abbildung 9

Im Modellexplorer des Innovators werden das Paket „Konzeptionelles Modell – Diagramme – Integrierte Datenschicht“ geöffnet und diejenigen Entity-Relationship-Diagramme ausgewählt, in denen die neuen Entitätstypen angelegt bzw. bestehende geändert werden müssen. Gegebenenfalls ist es notwendig ein zusätzliches Entity-Relationship-Diagramm für das neue Fachthema bzw. für die neuen Anforderungen zu erstellen. Änderungen in einem Entitätstyp müssen nicht unbedingt über das Diagramm, sondern können auch direkt über das Modellelement erfolgen.

Hinweis:

Es werden nur jene Elemente der HDS in die IDS übernommen, die zur Erfüllung der Anforderungen notwendig sind. Eine vollständige Abbildung der HDS in die IDS in einem Zuge erfolgt nicht. Stattdessen wird die IDS schrittweise bei der Umsetzung neuer Fachthemen oder Anforderungen ergänzt. Da die IDS als Data-Vault-Modell geführt wird, sind auch die Erweiterung einfach zu integrieren.

3.4.4.12 Prozessmodellierung von der HDS zur IDS

Die Abbildungsvorschriften, wie aus den Objekten der HDS die IDS-Ergänzungen durchgeführt werden, sind als Prozesse von der HDS zur IDS zu modellieren.

3.4.4.13 Erweiterung des DMS-Modells

Das DMS-Datenmodell wird um die neuen Anforderungen ergänzt. Dabei sind entweder bestehende Data Marts zu erweitern oder neue Data Marts zu erstellen.

Schritte im Modellierungswerkzeug:

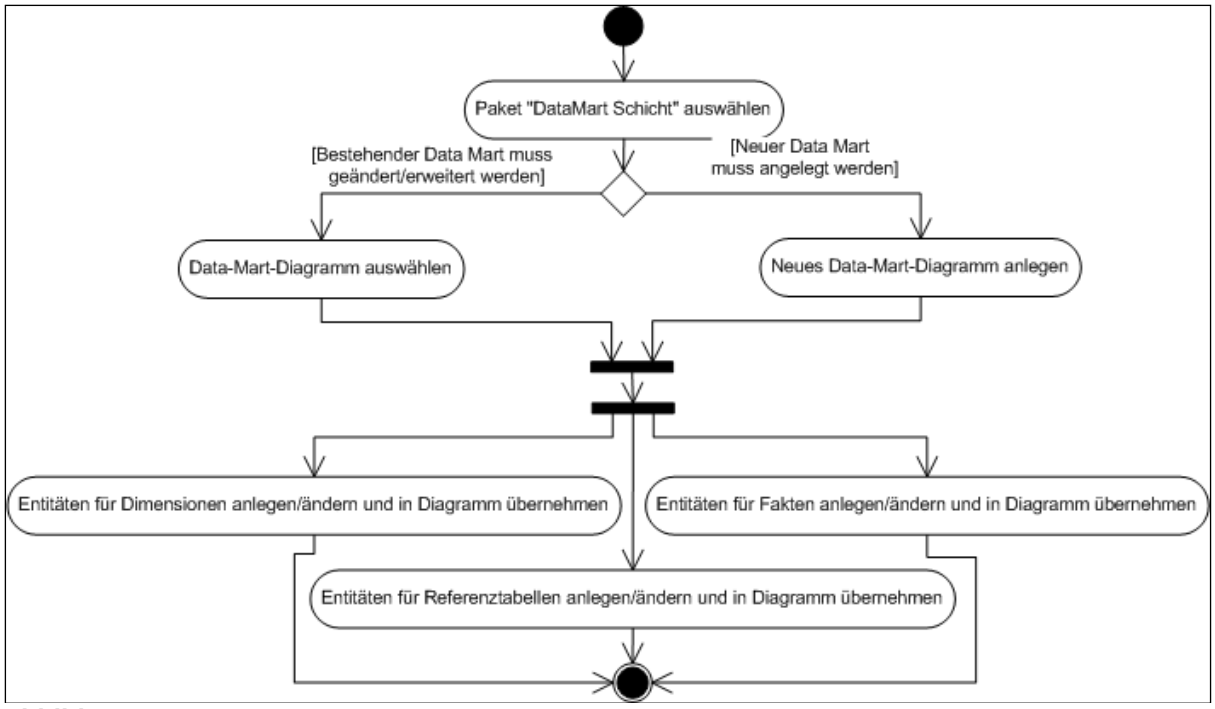

Abbildung 10

3.4.4.14 Prozessmodellierung von der IDS zur DMS

Die Abbildungsvorschriften, wie aus den Objekten der IDS die DMS-Ergänzungen durchgeführt werden, sind als Prozesse von der IDS zur DMS zu modellieren.

3.4.5 Leitfaden zur ETL-Entwicklung

Überblick:

1. Entwurf mit Modellierungswerkzeug erstellen.
2. Mapping Spezifikationen festlegen.
3. Erzeugen eines ETL-Jobs aus den Mapping-Spezifikationen.
4. Bearbeiten des erzeugten ETL-Jobs.
5. Reports und Analysen erzeugen

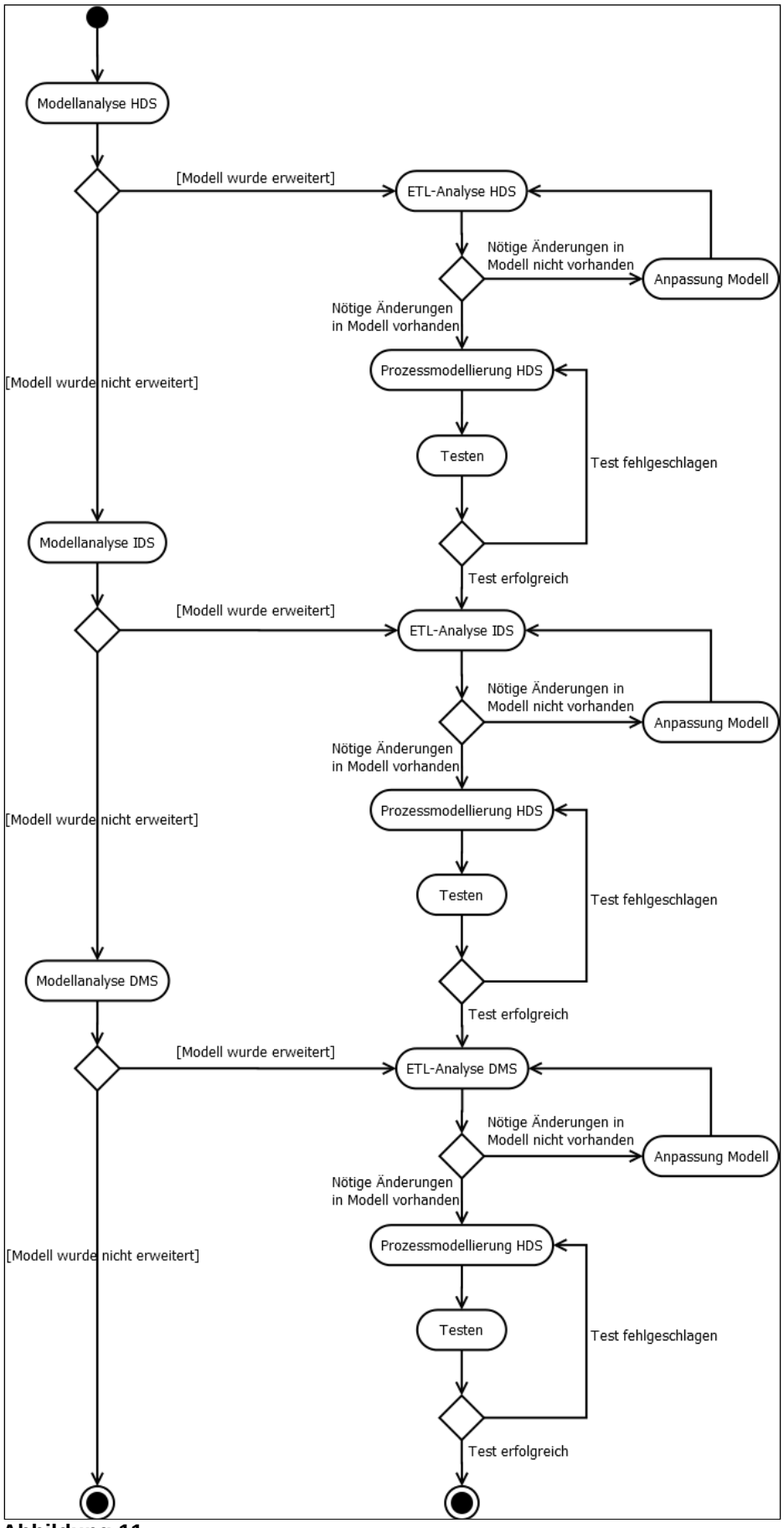

Abbildung 11

3.4.5.1 Modellanalyse HDS

Das Modell der HDS muss analysiert werden, um die getätigten Erweiterungen und Veränderungen festzustellen. Für die Veränderungen müssen bestehende ETL-Prozesse angepasst werden, für die Erweiterungen neue ETL-Prozesse erstellt werden.

Resultat:

- Prozessmodellierungen der veränderten und neuen HDS-Tabellen

3.4.5.2 Prozessanalyse HDS

Die Prozessmodellierungen müssen auf ihre Umsetzbarkeit geprüft und gegebenenfalls das Modell angepasst werden, bevor die Entwicklung startet.

Resultat:

- Liste Target-Tabellen mit jeweiligen Source-Tabellen
- Je Target-Tabelle Mapping Definition

3.4.5.3 Prozessmodellierung HDS

Die Metadateninformationen zu den Target- und Source-Tabellen müssen geladen und ggf. die Target-Tabellen mit einem generierten DDL-Script erstellt werden, wenn diese nicht vorhanden sind. Zwischen den Source- und Target-Tabellen werden die Mappingspezifikationen angewandt.

Skizze ETL:

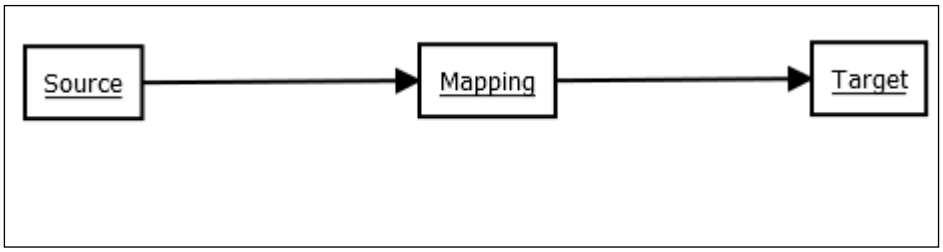

Abbildung 12

3.4.5.4 Modellanalyse IDS

Das Modell der IDS muss analysiert werden, um die getätigten Erweiterungen und Veränderungen festzustellen. Für die Veränderungen müssen bestehende ETL-Prozesse angepasst werden, für die Erweiterungen neue ETL-Prozesse erstellt werden.

Resultat:

- Prozessmodellierungen der veränderten und neuen IDS-Tabellen

3.4.5.5 Prozessanalyse IDS

Die Prozessmodellierungen müssen auf ihre Umsetzbarkeit geprüft und gegebenenfalls das Modell angepasst werden, bevor die Entwicklung startet.

Resultat:

- Liste Hub/Link/Satelliten-Tabellen mit jeweiligen Source-Tabellen
- Mapping Informationen mit JOIN und WHERE Bedingungen

3.4.5.6 Prozessmodellierung IDS

Anhand der Prozessdefinitionen des Modells müssen die ETLs der Hubs, Links und Satelliten modelliert werden. Die benötigten HDS-Tabellen werden ausgelesen, die nötigen Verarbeitungsschritte vorgenommen und in die zugehörige Hub-/Link-/Satelliten-Tabelle geschrieben. Die Beladungsreihenfolge ist wie folgt, wobei die Links gleichzeitig mit den Satelliten der Hubs geladen werden können.

3.4.5.6.1 Hub

Für die Beladung müssen die Business Keys der jeweiligen HDS-Tabelle ausgelesen und die Duplikate entfernt werden. Sind die BK nicht in der Hub Tabelle vorhanden, müssen sie mit einer generierten ID eingefügt werden.

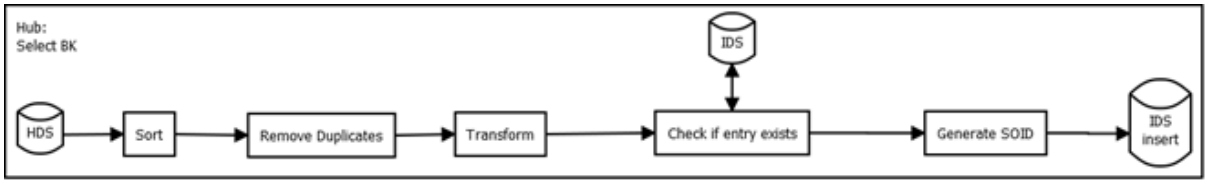

Abbildung 13

3.4.5.6.2 Link

Für die Beladung müssen die Business Keys der Hubs aus den HDS-Tabellen, welche die Verlinkungs-Informationen enthalten, ausgelesen und nach der Duplikat Entfernung durch die SOIDs aus den Hubs ersetzt werden. Die Kombination der Hub-SOIDs werden mit einer generierten SOID in den Link eingefügt, wenn die Kombination noch nicht existiert.

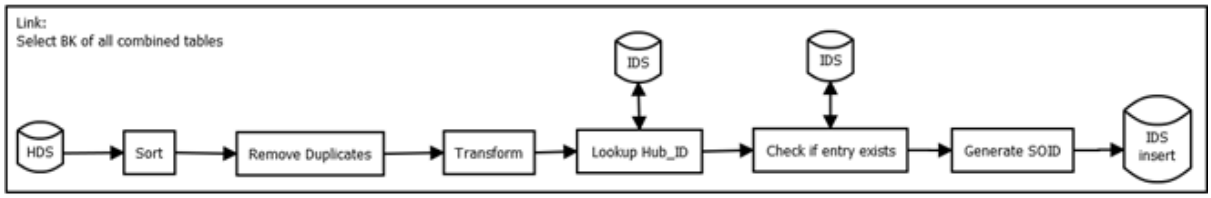

Abbildung 14

3.4.5.6.3 Satellit

Für den Satellit werden die benötigten HDS-Tabellen ausgelesen und die Business Keys durch die Hub-SOID ersetzt. Nach der Berechnung der technischen Gültigkeit wird überprüft, ob der Eintrag neu ist oder Veränderungen zum bestehenden Eintrag hat und gegebenenfalls eingefügt und vorhandene Einträge beendet.

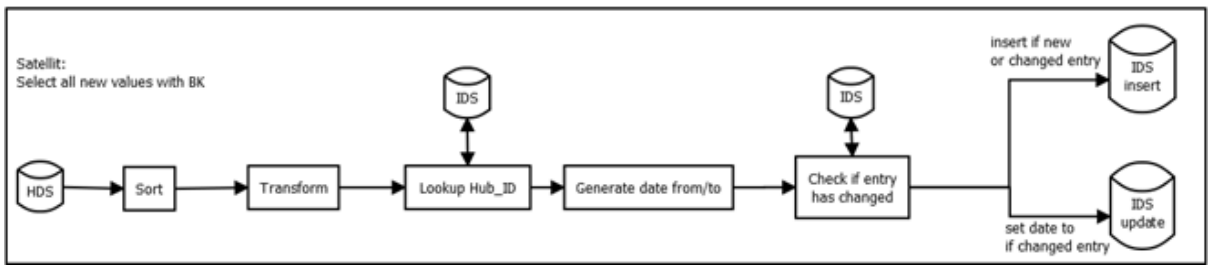

Abbildung 15

3.4.5.7 Modellanalyse DMS

Das Modell der DMS muss analysiert werden, um die getätigten Erweiterungen und Veränderungen festzustellen. Für die Veränderungen müssen bestehende ETL-Prozesse angepasst werden, für die Erweiterungen neue ETL-Prozesse erstellt werden.

Resultat:

- Prozessmodellierungen der veränderten und neuen DMS-Tabellen

3.4.5.8 Prozessanalyse DMS

Die Prozessmodellierungen müssen auf ihre Umsetzbarkeit geprüft und gegebenenfalls das Modell angepasst werden, bevor die Entwicklung startet.

Resultat:

- Liste Hub/Link/Satelliten-Tabellen mit jeweiligen Dimensions-/Fakten-Tabellen
- Mapping Informationen zum Laden der DIM/FACT-Tabellen und zeitlicher Stichtag

3.4.5.9 Prozessmodellierung DMS

Anhand der Prozessdefinitionen des Modells müssen die ETLs der Dimensionen und Fakten Tabellen modelliert werden. Die Hubs und Links mit zugehörigen Satelliten werden gefiltert gejoint und in die jeweilige DIM/FAKT-Tabelle geschrieben.

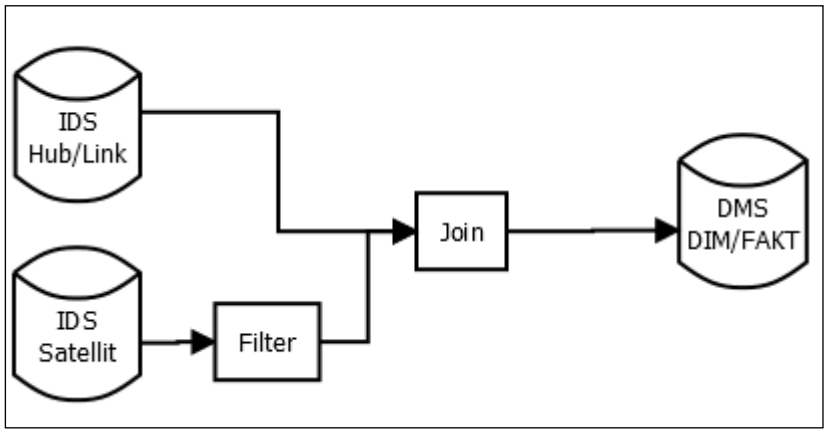

Abbildung 16

3.4.5.10 Mapping-Erstellung

Überblick:

1. Projekt erstellen
2. Metadaten importieren für Source und Target
3. Mapping-Spezifikationen
    a. Target
    b. Source
    c. Regeln
    d. Filter, Switches & Look-Up's
4. ETL- Job generieren

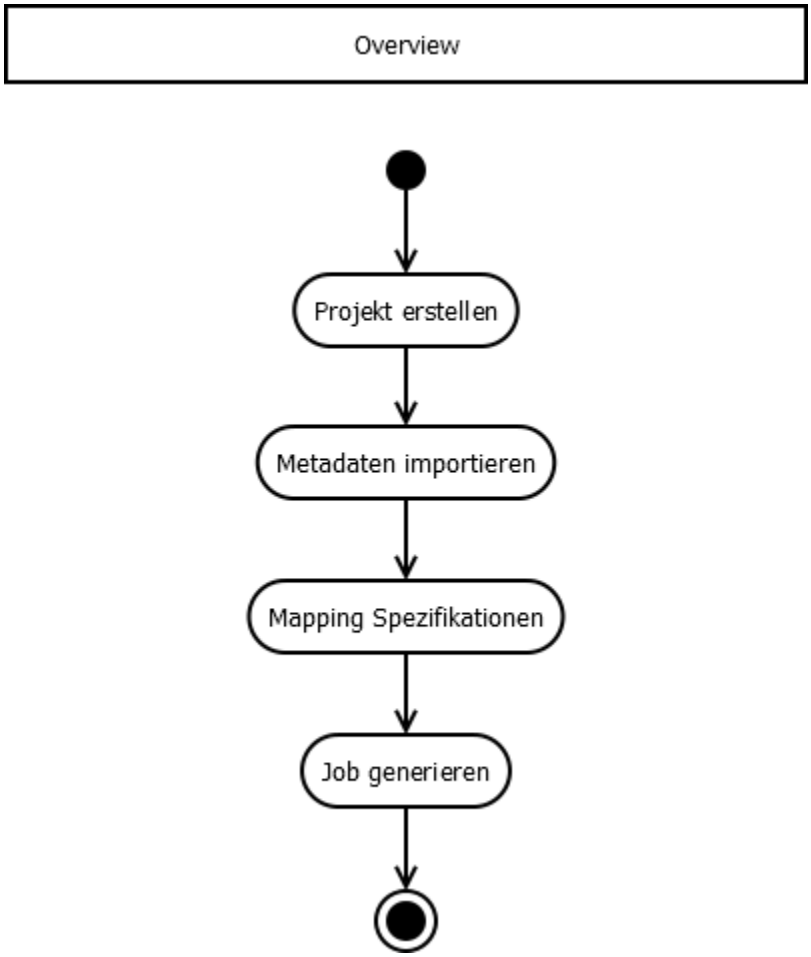

Abbildung 17

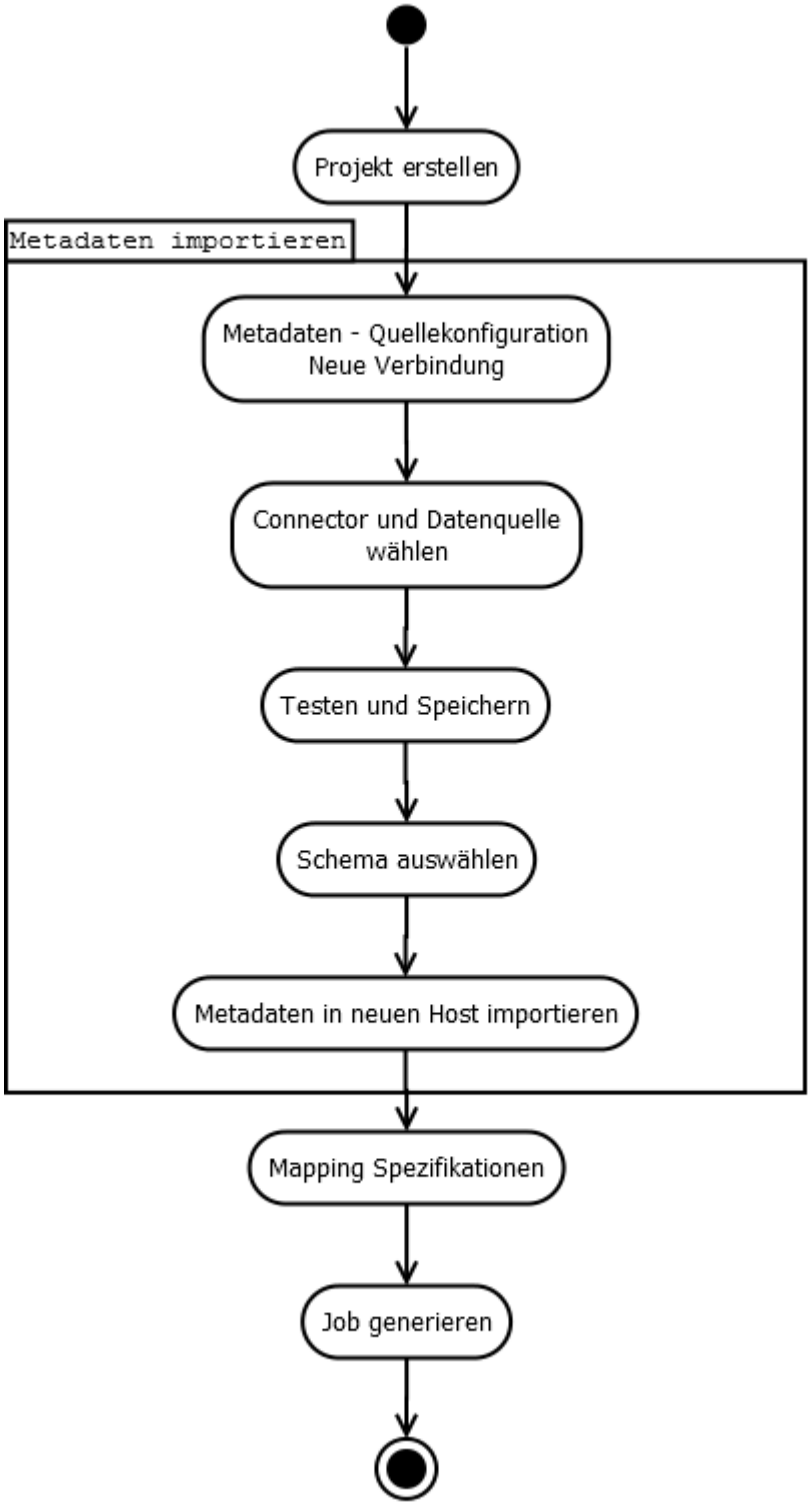

Abbildung 18

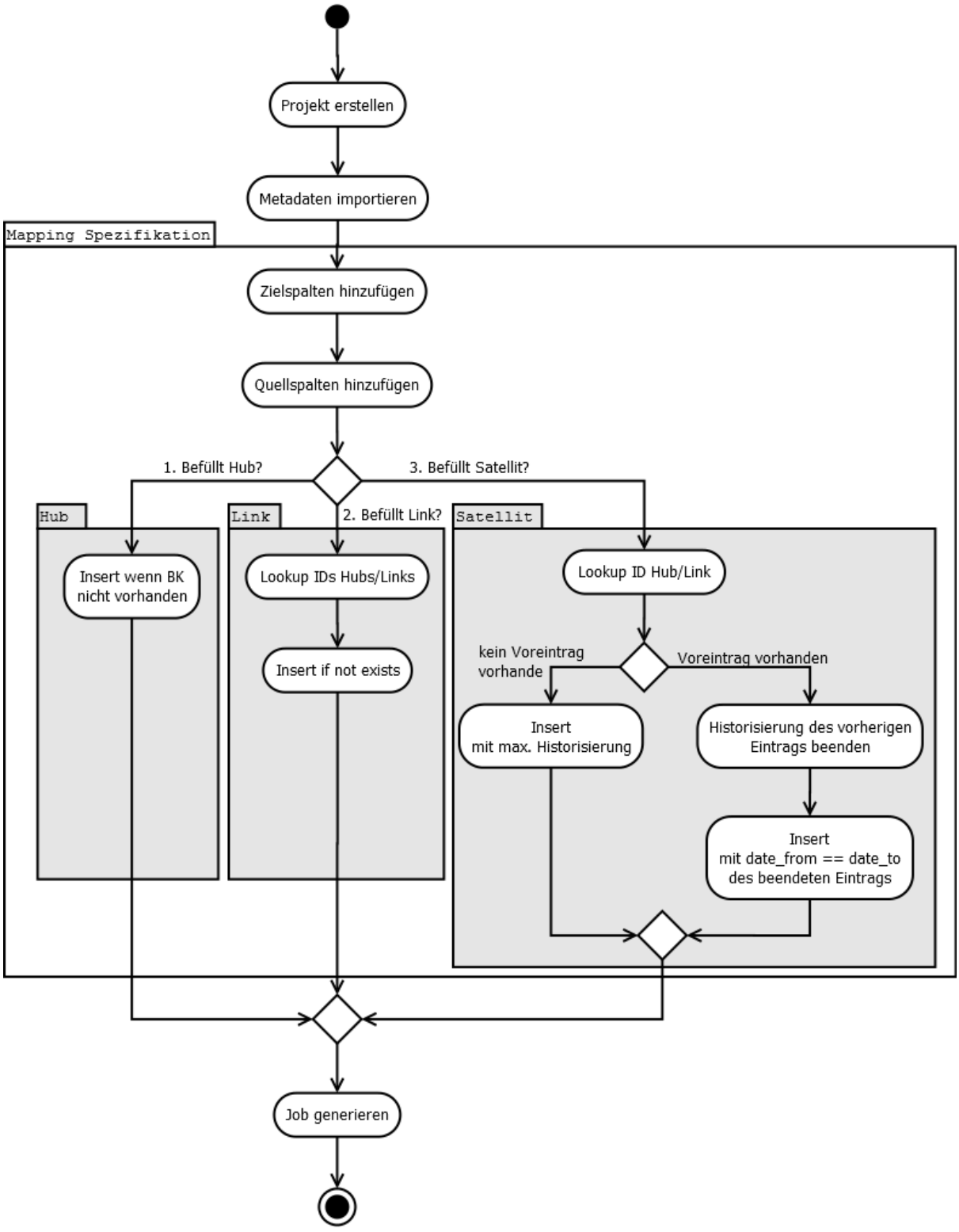

Abbildung 19

3.4.5.11 Versionierung, automatisiertes Testen und automatisierte QS

Ziel ist es die Teststellung zu automatisieren, damit im Entwicklungsteam keine speziellen administrativen Berechtigungen für einzelne Mitglieder vergeben werden müssen. Das komplette Thema befindet sich noch in Spezifikation.

4 Fazit

In der vorliegenden Darstellung wurde ein modellgetriebener Ansatz zur Einführung eines Vorgehensmodells bei der Entwicklung von BI- und DWH-Systemen vorgestellt. Als systematische Herangehensweise wurden dabei vier Projektdimensionen betrachtet: Managementsicht und Projektvorgehensweise, fachliche Sichtweise, Architektursicht und Modellsicht.

Während die fachlichen Inhalte (fachliche Sichtweise) und die Schichten einer BI- und DWH-Architektur in einem gewissen Sinn als „statisch“ vorgegeben anzusehen sind, stellen die Agilität und die Modellgetriebenheit die Kernelemente der Vorgehensweise selbst dar. Die Modellierung in den von der Architektur vorgegebenen Datenschichten (HDS, IDS, DMS) wurde dabei in drei Abstraktionsebenen (CIM, PIM, PSM) betrachtete und damit die Verbindung zu einer modellgetriebenen BI- und DWH-Entwicklung hergestellt.

5 Literatur

Balzert H. (1998). *Lehrbuch der Software-Technik. Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung.* Berlin: Spektrum Akademischer Verlag.

Golfarelli M., Maio D., Rizzi S. (1998). The dimensional fact model: A conceptual model for data warehouses. *Int. J. Cooperative Inf. Syst.*, 7(2-3):215–247.

Kimball R, Ross M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.).* Wiley.

Linstedt D. (2010). *Super Charge your Data Warehouse.*

Poole J. (2001). *Model-Driven Architecture: Vision, Standards And Emerging Technologies.*

Rodriguez A., Fernández-Medina E., Piattini M. (2007). *CIM to PIM transformation: A reality.* IFIP International Federation for Information Processing.

Schwaber K. (2007). *Agiles Projektmanagement mit Scrum.* Redmond: Microsoft Press.

Schwaber K. (2008). *Scrum im Unternehmen.* Redmond: Microsoft Pres.

Schwaber K., Beedle M. (2002). *Agile Software Development with Scrum.* Upper Saddle River: Prentice Hall.