



Bild: Shutterstock

**Wenn die Architektur Schritt für Schritt mitwächst**

## T-Shirt-Größen für BI-Lösungen

Der Chef braucht eine neue Auswertung auf Basis einer neuen Datenquelle. Die Daten zuerst in ein bestehendes geschweige denn in ein neues DWH zu laden ist zeitlich unrealistisch. Der Griff zum „agilen“ BI-Tool wie Excel oder Power BI liegt auf der Hand. Nach einigen Monaten und weiteren „dringenden“ Auswertungen droht der Wartungs Albtraum. Doch es geht auch anders: In diesem Artikel stelle ich die Idee von „Kleidergrößen“ für die Architektur vor, illustriert anhand eines eigenen Dashboards. Einerseits führen unterschiedliche Anforderungen zu unterschiedlichen Architekturansätzen, gleichzeitig soll jedoch die Architektur mitwachsen können, wenn ein BI-System nach und nach ausgebaut wird.

Ein Beitrag von  
Raphael Branger

Die Sommerferien waren gerade vorbei und ich stand kurz davor, eine neue Führungsfunktion innerhalb unseres Unternehmens zu übernehmen. Für mich war klar: Ich möchte meine Entscheidungen vorzugsweise datengetrieben fällen und auch meinen Mitarbeitern diese Möglichkeit anbieten. Ich hatte aus einem früheren Versuch bereits Kenntnis davon, wie ich die benötigten Daten aus unserem SAP-System extrahieren kann. Als BI-Experte hätte ich liebend gern gleich ein Data Warehouse (DWH) mit automatisierten Ladeprozessen etc. gebaut. Das Problem dabei: Als Kleinunternehmen hatten wir selbst keine wirkliche BI-Infrastruktur. Und meine eigene Zeit war – neben laufenden Kundenprojekten – ebenfalls sehr knapp bemessen. Ist das BI-Vorhaben damit gestorben? Natürlich nicht. Ich bediente mich einfach derjeni-

gen Hilfsmittel, die ich gerade zur Hand hatte. In diesem Fall war es Microsoft Power BI. In wenigen Stunden stand bereits das erste Dashboard, und noch schneller war es im Cloud-Service publiziert.

### Die Nachteile von Quick & Dirty

Kurzfristig war mein Problem also gelöst. Den Mitarbeitern konnte ich tagesaktuelle Daten zur Verfügung stellen. Weitere Anforderungen folgten. Also kopierte ich die Power-BI-Datei und begann sie da und dort anzupassen. Im Laufe der Wochen kamen immer wieder neue Kennzahlen und weitere Kopien dazu. Die verschiedenen Dashboard-Kopien „in sync“ zu halten gelang mir nur bedingt. Dazu kamen operative Probleme: Natürlich verband ich mich mit dem SAP-System unter meinem persönli-

Sonderdruck aus  
BI-SPEKTRUM 3/2019

chen Benutzernamen – dessen Passwort regelmäßig geändert werden musste. Das wiederum führte zu Unterbrüchen bei der Datenaktualisierung und manuellem Wartungsaufwand für die Nachkonfiguration des neuen Passworts in Power BI.

### Stufenweise Entwicklung der BI-Architektur

So wie mir geht es wohl vielen Fachanwendern. Auf die Schnelle muss eine Lösung her – und schneller, als man denkt, befindet man sich im persönlichen Wartungsalltraum. Gleichzeitig liegt eine ausgewachsene BI-Lösung in weiter Ferne, meist aus organisatorischen oder finanziellen Gründen. Als Experte in der Adaption agiler Methoden für BI-Projekte beschäftige ich mich schon lange mit dieser Problemstellung: Wie kriegen wir die kurzfristigen Bedürfnisse der Fachanwender mit der Nachhaltigkeit einer „sauberen“ Lösung unter einen Hut?

Als Vater zweier Töchter erinnerte ich mich an den Wachstumsprozess meiner Kinder. Dazu gehört auch der Umstand, dass man regelmäßig größere Kleidung kaufen muss. Der Wachstumsprozess erfolgt kontinuierlich – von Zeit zu Zeit muss jedoch eine neue Kleidergröße her. Genau dieses Prinzip lässt sich auch auf BI-Lösungen beziehungsweise deren Architektur anwenden. In Abbildung 1 sind vier exemplarische Größen abgebildet, die im Folgenden näher erläutert werden.

#### Think Big - Start Small: Größe S

Dieser Ansatz entspricht meinem ersten Wurf des Dashboards. Ein BI-Frontend-Werkzeug verbindet sich direkt auf eine Quelle. Sämtliche Metadaten wie

zum Beispiel Zugriffsdaten für die Quellsysteme, Geschäftsregeln, Kennzahldefinitionen etc. werden direkt im Kontext der erstellten Informationsprodukte entwickelt und gespeichert. Diese Größe ist geeignet, wenn sich ein BI-Vorhaben noch ganz am Anfang befindet. Oft wird dann ein großes Ziel formuliert, was man alles analysieren und auswerten möchte – das ist der Teil „Think Big“. Praktisch ist aber nicht viel mehr bekannt als die Datenquelle. Dann möchte man eher explorativ und zeitnah erste Ergebnisse produzieren – hier macht es Sinn, klein und technisch wenig anspruchsvoll zu starten.

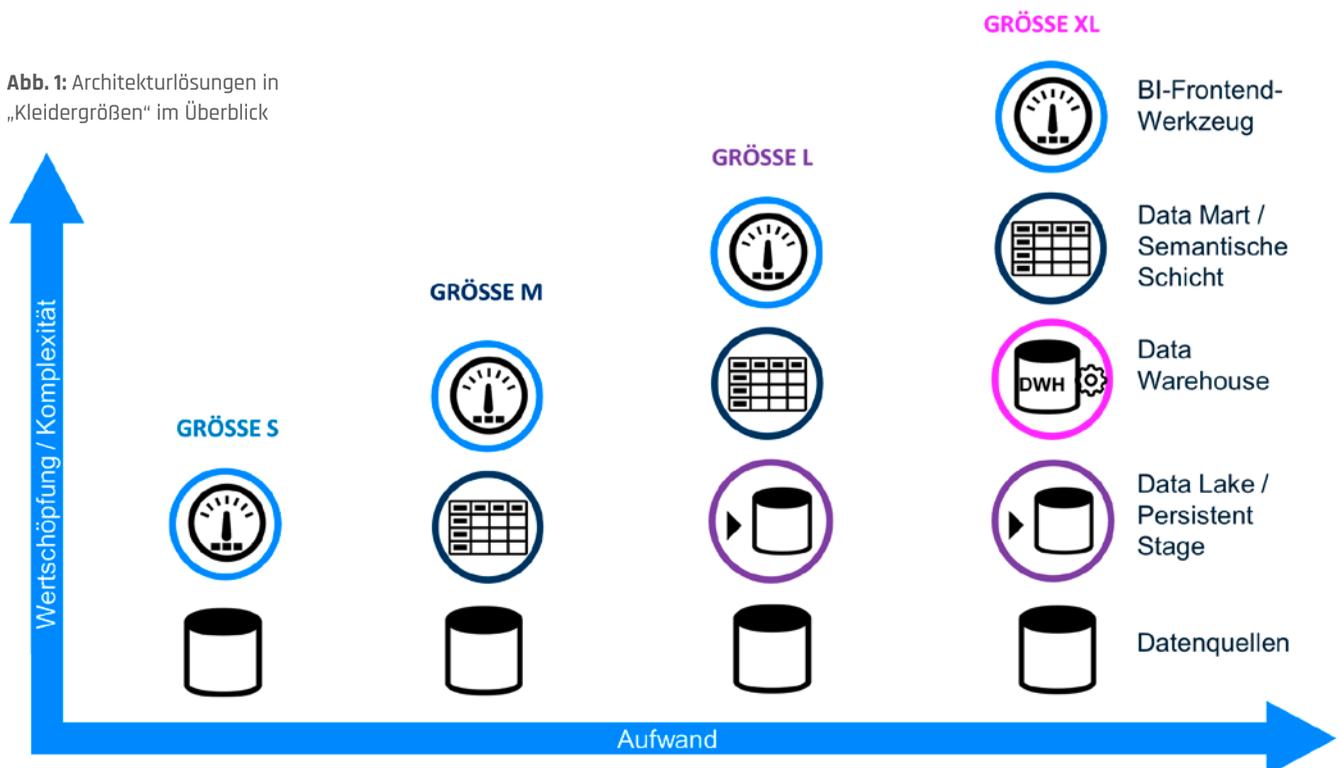
Dieser Ansatz kommt jedoch rasch an seine Grenzen. Im Folgenden eine nicht abschließende Liste möglicher Kriterien, um zu ermitteln, wann es an der Zeit ist, die nächste architektonische Kleidergröße in Betracht zu ziehen:

- Es existieren mehrere ähnliche Informationsprodukte beziehungsweise Datenabfragen, welche die immer wieder gleichen Kennzahlen und Attribute verwenden.
- Es greifen verschiedene Benutzer regelmäßig auf die Informationsprodukte zu – der Zugriff auf die Datenquelle erfolgt jedoch mit dem persönlichen Zugriffskonto des ursprünglichen Entwicklers.
- Das Quellsystem leidet unter den mehrfachen (meist sehr ähnlichen) Datenabfragen der verschiedenen Informationsprodukte.

#### Eine gemeinsame Sprache entwickeln - Größe M

Auf der nächsten Stufe versuchen wir, gemeinsam nutzbare Metadaten wie Kennzahldefinitionen oder die Zugriffsinformationen für die Quellsysteme in einer eigenen Ebene – statt dem Informati-

Abb. 1: Architekturlösungen in „Kleidergrößen“ im Überblick



onsprodukt selbst – zu speichern. Diese Ebene wird häufig auch als Data Mart oder semantischer Layer bezeichnet. Im Falle meines eigenen Dashboards entwickelten wir dafür ein tabulares Modell in Azure Analysis Services (AAS). Die verschiedenen Ausprägungen beziehungsweise Kopien des Dashboards als solches blieben dabei zu großen Teilen erhalten – lediglich der Unterbau veränderte sich. Alle Varianten zeigten nun aber auf die gleiche, zentrale Grundlage. Die Vorteile dieser Größe im Vergleich zur vorherigen liegen auf der Hand: Ihr Wartungsaufwand wird erheblich reduziert, weil die Datengrundlage einmal zentral und nicht für jedes einzelne Informationsprodukt gepflegt werden muss. Gleichzeitig bringen Sie Konsistenz in die Kennzahldefinition sowie die Benennung der Kennzahlen. In einem mehrsprachigen Umfeld tritt der Mehrwert noch stärker zutage, weil Übersetzungen nur einmal zentral und einheitlich im semantischen Layer gepflegt werden müssen. Alle Dashboards sprechen dadurch eine gemeinsame Sprache.

In dieser Größe M speichern wir noch keine Daten dauerhaft außerhalb der Quelle. Selbst wenn die Quelldaten ins tabulare Modell in AAS importiert werden, müssen diese bei größeren Anpassungen am Modell immer wieder neu geladen werden. Andere Hersteller kommen ganz ohne Datenspeicherung aus, zum Beispiel die sogenannten Universen in SAP BusinessObjects. Das bedeutet im Umkehrschluss eine teils hohe Last auf den Quellsystemen insbesondere während der Entwicklungsphase.

Im Folgenden wiederum eine Liste der möglichen Gründe, Ihrem „BI-Kind“ die nächstgrößere Kleidergröße zu spendieren:

- Die Abfragebelastung der Quellsysteme ist trotz eines semantischen Layers noch zu groß und soll weiter reduziert werden.
- Die BI-Lösung soll auch als Archiv für die Quelldaten dienen, zum Beispiel im Falle von web-basierten Datenquellen, wo die Historie nur für einen begrenzten Zeitraum zur Verfügung steht.
- Wenn das Quellsystem selbst keine Änderungen an Datensätzen historisiert, ist dies ein weiterer Beweggrund, die BI-Lösung als Datenarchiv auszuliegen.

## Entkopplung von der Quelle – Größe L

Die nächstgrößere Größe L für unsere BI-Architektur ersetzt den direkten Datenzugriff des Data Mart auf die Quelle. Dafür werden die Daten aus der Quelle extrahiert und permanent in einer separaten Datenbank gespeichert. Diese Ebene entspricht den Konzepten einer Persistent Staging Area (PSA) (vgl. dazu [Kim04] S. 31f. sowie [Vos16]) oder eines aktiv verwalteten Data Lake (vgl. dazu [Gor19], zum Beispiel S. 9). Allen gemeinsam ist, dass die Daten möglichst unverändert aus der Quelle übertragen und dauerhaft gespeichert werden. Dieses Vorgehen führt dazu, dass die bereits bestehenden Data Marts relativ einfach auf diese neue Quelle umgehängt werden können.

**RAPHAEL BRANGER** studierte Wirtschaftsinformatik und hält einen MA in Information Management. Heute arbeitet er als Senior Solution Architect bei der IT-Logix AG mit über 17 Jahren Praxiserfahrung im Business-Intelligence- und Data-Warehousing-Umfeld. Sein aktueller Fokus liegt im Bereich der BI-spezifischen Anforderungsanalyse sowie der Adaption agiler Methoden im Umfeld von BI.

**E-Mail:** [rbranger@it-logix.ch](mailto:rbranger@it-logix.ch)



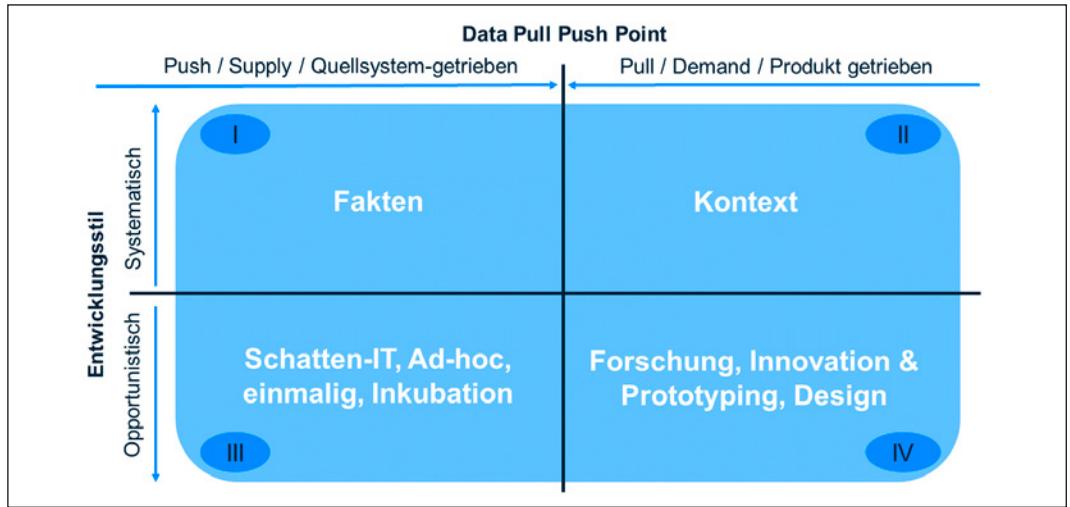
Für mein eigenes Dashboard-Beispiel nutzen wir diese Stufe zurzeit noch nicht. Wir haben aber geplant, im nächsten Schritt die SAP-Daten mittels Azure Data Factory zu extrahieren und dauerhaft in einer Azure-SQL-Datenbank abzuspeichern. Gerade wenn man – wie wir – das ERP als Cloud-Lösung nutzt, reduziert dieser Layer den „Lock-in“-Effekt gegenüber dem ERP-Cloud-Anbieter. Weitere Vorteile dieser persistenten Datenspeicherung außerhalb der Quellsysteme ist die damit einhergehende Archiv- und Historisierungsfunktion: Sowohl neue als auch geänderte Daten aus der Quelle werden kontinuierlich gespeichert. Gelöschte Datensätze können entsprechend markiert werden. Datenmodell-technisch bewegen wir uns sehr nahe am Datenmodell der jeweiligen Quelle, unter Umständen harmonisieren wir bereits einige Datentypen. So praktisch und rasch umsetzbar dieser Layer ist, auch hier gibt es Indizien, wann der Sprung zur nächsten Größe angebracht ist:

- Die gewünschten Informationsprodukte benötigen integrierte und harmonisierte Daten aus mehreren Datenquellen.
- Die Datenqualität der Quelldaten ist nicht ausreichend und kann auch nicht einfach im Quellsystem verbessert werden.
- Berechnete Kennzahlen sollen dauerhaft gespeichert werden, zum Beispiel zu Auditzwecken.

## Integrierte, harmonisierte und historisierte Daten – Größe XL

Die nächste und vorerst letzte Größe XL erweitert die bestehende Architektur um eine klassische Data-Warehouse-Struktur zwischen PSA und dem Data Mart. Dabei steht die Integration und Harmonisierung von (Stamm-)Daten aus den verschiedenen Quellsystemen im Vordergrund. Dies erfolgt über ein zentrales Datenmodell, das unabhängig von der verwendeten Quelle existiert (zum Beispiel ein Data-Vault-Modell oder ein dimensionales Modell). Damit verbunden sind auch systematische Datenqualitätskontrollen bereits beim Laden und Verarbeiten der Daten. Ebenso können hier nach Bedarf Historisierungskonzepte aufbauend auf die integrierten Daten umgesetzt werden. Durch die persistente Speicherung der Daten in diesem DWH-Layer stehen diese auch für Auditzwecke dauerhaft zur Verfügung.

Abb. 2: Das Data Quadrant Model nach Ronald Damhof



Die verschiedenen Größen unterscheiden sich aber nicht nur in der Anzahl der eingesetzten Ebenen. Um die vier Architekturansätze umfassender charakterisieren zu können, soll hier kurz das Modell der Data Management Quadrants nach [Dam 15] eingeführt werden.

**Die verschiedenen Größen im Kontext der „Data Management Quadrants“**

In [Dam 15] beschreibt Ronald Damhof ein einfaches Modell zur Positionierung von Daten-Management-Vorhaben in einer Organisation. Dabei unterscheidet er zwei Dimensionen und vier durchnummerierte Quadranten (vgl. dazu auch Abbildung 2).

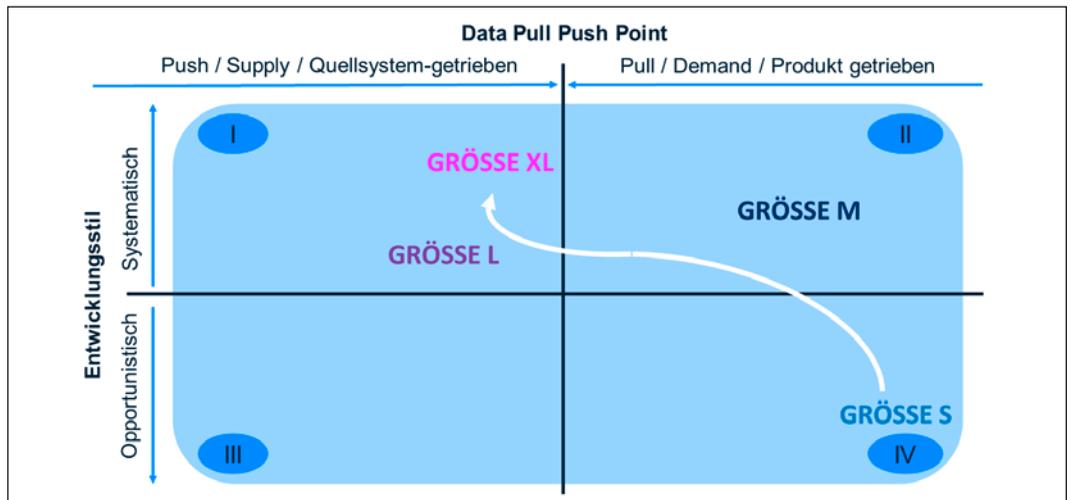
Damhof zieht auf der x-Achse die aus der Betriebswirtschaftslehre bekannten Begriffe der Push- oder Pull-Strategie heran. Gemeint ist damit, wie stark der Produktionsprozess durch die Nachfrage gesteuert und individualisiert wird. Auf der rechten beziehungsweise Pull-Seite werden auf Basis ganz konkreter Fachanforderungen zum Beispiel themenspezifische Datamarts und darauf aufbauend Informationsprodukte wie Reports und Dashboards entwickelt. Hier stehen Agilität und Fachwissen im Zentrum. Die ersten beiden Größen S und M können auf dieser Seite verordnet werden.

Auf der anderen (linken beziehungsweise Push-) Seite stellt zum Beispiel die BI-Abteilung die Anbindung verschiedener Quellsysteme zur Verfügung und bereitet die Daten in einem Data Warehouse auf. Der Fokus liegt auf Skaleneffekten und dem Bereitstellen einer stabilen Grundinfrastruktur für BI im Unternehmen. Hier können wir die beiden anderen Größen L und XL verorten.

Auf der y-Achse steht bei Damhof die Art und Weise, wie ein Informationssystem oder -produkt „produziert“ wird. In der unteren Hälfte erfolgt die Entwicklung opportunistisch. Entwickler und Nutzer sind hier häufig identisch. Beispielsweise wird eine aktuelle Problemstellung mit Daten in Excel oder mit anderen Tools direkt durch den Fachanwender ausgewertet. Dies entspricht der ersten Größe S. Wie bereits in meinem eigenen Fallbeispiel ersichtlich, geht die Flexibilität, die etwa für Forschung, Innovation oder Prototyping gewonnen wird, auf Kosten der Einheitlichkeit und Wartbarkeit der Resultate. Verlässt der entsprechende Fachanwender das Unternehmen, geht häufig auch das Wissen über die Analyse und die angewandten Geschäftsregeln verloren.

Im Gegensatz dazu erfolgt die Entwicklung in der oberen Hälfte systematisch: Hier sind Entwickler und Endbenutzer typischerweise unterschiedliche Personen. Die Datenbeschaffungsprozesse

Abb. 3: Die Größen im Kontext des Data Quadrant Model



sind weitestgehend automatisiert und hängen daher im täglichen Betrieb nicht von der Gegenwart einer bestimmten Person ab. Die Daten haben eine hohe Zuverlässigkeit infolge systematischer Qualitätssicherung, Kennzahlen sind einheitlich definiert. Die Aspekte der Größen L und XL sind in den meisten Fällen hier zu finden.

Die verbleibende Größe M befindet sich irgendwo auf „dem Weg“ zwischen Quadrant IV und II. So ist es sicherlich auch einem Fachanwender ohne IT-Support möglich, den Data Mart umzusetzen. Wird die Lösung zudem auch hinsichtlich Entwicklung und Betrieb systematisiert, kann sich dieser Ansatz auch im zweiten Quadranten befinden. Das zeigt zudem, dass die Architektur-Größen nicht nur hinsichtlich der Anzahl eingesetzter Ebenen wachsen.

Die Positionierung der verschiedenen Größen im Quadranten-Modell (vgl. dazu Abbildung 3) drückt zwei weitere Bewegungen aus.

- Die Bewegung von unten nach oben: Wir erhöhen die Systematisierung, indem die Lösung unabhängig(er) vom ursprünglichen Fachanwender wird. In meinem eigenen Dashboard drückte sich das zum Beispiel dadurch aus, dass der Datenzugriff irgendwann nicht mehr mit meinem persönlichen SAP-Benutzernamen erfolgte, sondern mittels eines technischen Kontos. Ein anderer Aspekt der Systematisierung ist die Anwendung der Datenmodellierung. Während mein initiales Dashboard einfach eine breite Tabelle importierte, wurden im tabularen Modell die Daten bereits dimensional modelliert.
- Die Bewegung von rechts nach links: Während die ersten beiden Größen klar dominiert sind durch die fachlichen Anforderungen und entsprechendes Domänenwissen, sind weiter links mehr und mehr technische Fertigkeiten gefragt, zum Beispiel um unterschiedliche Datenformate und -typen zu handhaben oder Prozesse zu automatisieren.

## Literatur

[Dam15] Damhof, R.: „Make data management a live issue for discussion throughout the organization“. 15.6.2015, <https://prudenza.typepad.com/files/english---the-data-quadrant-model-interview-ronald-damhof.pdf>, abgerufen am 29.7.2019

[Gor19] Gorelik, A.: The Enterprise Big Data Lake. O'Reilly 2019

[Kim04] Kimball, R.: The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data. Wiley 2004

[Vos16] Vos, R.: „Why you really want a Persistent Staging Area in your Data Vault architecture“. 25.6.2016, <http://roelantvos.com/blog/why-you-really-want-a-persistent-staging-area-in-your-data-vault-architecture/>, abgerufen am 29.7.2019

## Zusammenfassung und Ausblick

Fassen wir zusammen: BI-Lösungen müssen mit den Anforderungen mitwachsen. Die gezeigten Architekturlösungen in vier Größen illustrieren exemplarisch, wie dieser Wachstumspfad konkret aussehen könnte. Dabei wird die DWH-Lösung sozusagen von oben nach unten gebaut – wir starten mit dem reinen Informationsprodukt und bauen danach die Datengrundlage Schritt für Schritt aus bis zur vollständigen Data-Warehouse-Architektur.

Die verschiedenen Architekturansätze lassen sich auch im Quadranten-Modell von Ronald Damhof positionieren: Eine neue BI-Lösung entsteht oft im vierten Quadranten, wo Fachanwender explorativ mit Daten arbeiten und erste Versionen von Informationsprodukten schaffen. Bewähren sich diese, gilt es insbesondere deren Datengrundlage zu systematisieren und zu vereinheitlichen. In einem ersten Schritt dient ein Data Mart als Garant für eine von verschiedenen Informationsprodukten genutzte Sprache. Eine datenmäßige Entkopplung von den Quellsystemen erlaubt zudem eine weitere Skalierung der Entwicklungsarbeiten. Zu guter Letzt können die bisherigen Ebenen um ein Data Warehouse ergänzt werden, um Daten aus verschiedenen Quellen dauerhaft zusammenzuführen und bei Bedarf historisiert zur Verfügung zu stellen.

Organisationen sollten es sich zum Ziel setzen, den Wachstumsprozess einer BI-Lösung zu institutionalisieren. Fachanwender können nicht bei jeder neuen Datenquelle darauf warten, dass diese zunächst über verschiedene Schichten integriert und erst dann für Auswertungen bereitgestellt wird. Andererseits müssen individuelle Lösungen laufend systematisiert und nach und nach auf ein stabiles (Daten-)Fundament gestellt und ordentlich betrieben werden. Die gezeigten Architekturansätze in Größen oder Stufen geben Anhaltspunkte, wie diese Institutionalisierung aussehen könnte.

# IT-LOGIX

BUSINESS INTELLIGENCE



« LET YOUR DATA SPEAK STEP BY STEP »

[www.it-logix.ch](http://www.it-logix.ch)