



□ Albert Wilhelm

(E-Mail: albert.wilhelm@de.ibm.com)
 ist für die technische Kundenbetreuung der IBM Deutschland GmbH (www.ibm.de) rund um das Thema Rational System Architect zuständig. Er hat mehrjährige Erfahrung in der Prozessanalyse und im Anforderungsmanagement und beschäftigt sich im Rahmen seiner Tätigkeit intensiv mit der Konzeption, Entwicklung und Anwendung von Architekturmodellen.



□ Hubert Zenner

(E-Mail: hubert.zenner@de.ibm.com)
 unterstützt den Vertrieb der IBM Deutschland GmbH (www.ibm.de) in allen technischen Fragen rund um Rational System Architect. Seine langjährigen Erfahrungen im Softwarevertrieb, in unterschiedlichen Branchen und diversen Projekten bilden die ideale Basis für seine aktuelle Arbeit im Umfeld von Architektur- und Prozessmanagement.

Vom Häuslebau und Informationsverhau

Wenn es um Software-Architektur und ihre Stellung in der Entwicklungs- und Wartungsphase geht, dient des Öfteren ein Vergleich mit Gebäudearchitektur zur Veranschaulichung und zum besseren Verständnis der Zusammenhänge. Diese Parallele wirkt deshalb so anschaulich, weil jeder persönliche Erfahrungen mit Gebäudearchitektur gemacht hat – sei es als Nutzer (zu Hause, im Büro, in öffentlichen Einrichtungen) oder als Gestalter (bei der Wohnungssuche, beim Heimwerken oder dem Hausbau). Nutzt man diesen Erfahrungsschatz, so lassen sich viele Begriffe, Zusammenhänge und Abstraktionen einer Software-Architektur klarer veranschaulichen. Doch die meisten dieser Begriffe sind technischer Natur mit Fokus auf Details und Strukturen der Gebäude, aber ein wohlgesonnener Immobilienmakler wird Ihnen bestätigen, dass es neben den baulichen und technischen Merkmalen einer Immobilie noch drei Faktoren gibt, die ihren Wert und den Werterhalt maßgeblich bestimmen: die Lage, die Lage und die Lage. Bei der großen Ähnlichkeit zwischen Gebäude- und Software-Architektur stellt sich nun die Frage, ob diese Weisheit aus der realen Welt eine Parallele in der Software-Welt hat, und welche Konsequenzen man daraus ziehen kann.

Tatsächlich lässt sich die *Lage* einer Software bzw. eines Entwicklungsprojektes mithilfe geeigneter Mittel überwachen und bewerten. Solche Maßnahmen tragen sogar erheblich zum Erfolg einer Unternehmung bei. Die Notwendigkeit, Techniken und der Nutzen der aktiven Überwachung von Einflussfaktoren werden auf den folgenden Seiten erläutert.

Der subjektive Begriff „Lage“ fasst in diesem Zusammenhang bewertete Einflussfaktoren, die sich von uns kaum oder gar nicht manipulieren lassen, die aber sehr wohl den Erfolg der Unternehmung mitbestimmen. G. Starke unterscheidet in seinem Buch *Effektive Software-Architekturen* [Sta02] drei Kategorien von Einflussfaktoren:

- organisatorische und politische Faktoren, die sog. „facts of life“;
- technische Faktoren;
- System- oder Produktfaktoren.

Als Software-Entwickler hat man kaum die Möglichkeit auf diese Einflussfaktoren ein-

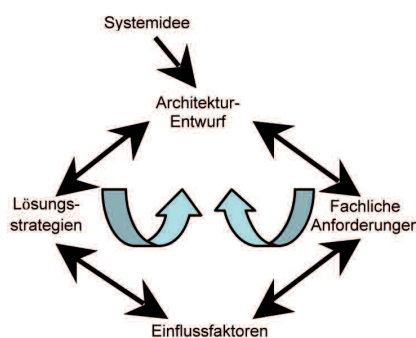


Abb. 1: Gegenseitige Beeinflussung von Entwurf, Lösungs- und Einflussfaktoren nach Starke [Sta02]

zuwirken, sie dagegen haben direkten Einfluss auf jedes nicht triviale System, ähnlich der Lage bei einer Immobilie. Doch während sich die Lage der Immobilie im Laufe der Zeit nur wenig ändert – Autobahnen, Flughäfen, Schulen werden nicht jedes Jahr in der Nachbarschaft gebaut – ist die *Lage* der Software im steti-

gen Wandel: Anforderungen, Märkte und Technologien verändern sich. Geschäftsziele, die in einem Quartal auf die Fahnen geschrieben wurden, sind im nächsten bereits obsolet, Unternehmen fusionieren, werden restrukturiert, neu orientiert usw. Der Einfluss dieser Änderungsprozesse ist umfassend, auch wenn er sich auf einzelne Algorithmen weniger stark auswirkt als auf komplexe Systeme. Starke beschreibt die gegenseitige Beeinflussung von Entwurf, Lösungs- und Einflussfaktoren wie in **Abbildung 1**.

Im Folgenden sind einige Beispiele für die o.g. Einflussfaktoren zu finden.

Die Ziele einzelner Stakeholder, die durchaus verschieden sein können, gehören ebenso wie die Unternehmens- und Geschäftsziele zu den o.g. *politischen Faktoren*. Sie müssen bekannt sein und wachsam verfolgt werden – wenn nicht deutlich gemacht wird, dass mit der Software die geltenden Ziele unterstützt werden, oder das Entwicklerteam auf einen Richtungswechsel nicht rechtzeitig reagiert,

dann besteht die Gefahr, dass das Produkt trotz aller technischer Feinheiten nicht zum Einsatz kommt. E. Rehtin beschreibt in seinem Buch *The Art of Systems Architecting* den Zusammenhang wie folgt: „Es gibt keine rein technischen Probleme. Sie werden schnell zu organisatorischen oder politischen Schwierigkeiten, und damit entgleiten sie auch der Kontrolle rein technisch orientierter Projektbeteiligter.“

Die bestehende technologische Systemlandschaft und technologischer Fortschritt gehören zu den o.g. *technischen Faktoren* und bilden ebenfalls wichtige Rahmenbedingungen, die bekannt und überwacht werden sollten. Entwickelt sich z. B. ein Datenbankmanagementsystem zum Standard, welches unser Produkt nicht unterstützt und nicht auf der Roadmap hat, dann wird dieser Umstand sich eher früher als später negativ bemerkbar machen.

Funktionale und nichtfunktionale Anforderungen an das System bilden die o.g. *System- oder Produktfaktoren*, welche von den Entwicklern oft fälschlicherweise als seine einzigen Einflussfaktoren wahrgenommen werden, sei es aus Bequemlichkeit oder Spezialisierung. Dieser Fokus auf rein funktionelle Aspekte des Produkts verhindert eine realistische Beurteilung der momentanen und bevorstehenden Lage. Ch. Bommer fasst es in seinem Buch *Softwarewartung* treffend zusammen: „Wir müssen die sich ständig verändernde Umgebung aktiv wahrnehmen und gezielte Anpassungen unseres Systems daran in die nächste Release einplanen und umsetzen. Oder kurz gesagt: wir müssen laufende Veränderungen bewusst managen“ [Bom08].

Informationen sammeln

In der Software-Architektur spricht man im Zusammenhang mit all diesen Einflussfaktoren von der Unternehmensarchitektur oder Enterprise Architecture (EA). Die Informationen dazu finden sich in vielen unterschiedlichen Systemen, Tabellen, Dokumenten oder aber auch nur in den Köpfen der Mitarbeiter. Wollen wir nun die sich ständig ändernde Umgebung aktiv wahrnehmen, müssen wir eine Plattform schaffen, mit der dies möglich wird. Verteilte Informationen in all den unterschiedlichen Formaten und Werkzeugen sind hier nicht dienlich, da sie immer nur eine Teilwahrheit darstellen, aber nicht den Blick auf die übergeordneten Zusammenhänge freigeben. Die Zusammenführung in ein zentrales System bringt hier entspre-

chenden Mehrwert und hilft dabei, all die Informationen in nutzbringendes Wissen zu verwandeln. Ein solches zentrales Informationssystem für Unternehmensarchitekturen stellt IBM Rational System Architect dar. In einem solchen System sammeln sich Informationen wie etwa:

- Unternehmensziele und Visionen
- Strategien und Geschäftsregeln
- Prozesse und Anwendungen
- Datenmodelle der Datenbanken
- Hardware- und Infrastrukturinformationen.

Dabei wird das System für viele Informationen zur neuen Heimat werden, manche Informationen werden jedoch mit anderen Werkzeugen synchronisiert werden. Ein Beispiel hierfür sind CMDB-Werkzeuge wie Tivoli, die sehr viele Informationen direkt aus dem operativen Umfeld sammeln. Sie sind ein nützlicher Partner für die Erfassung der Ist-Architektur. Für die Planungsaufgaben in der Unternehmensarchitektur werden davon nur Teile oder Aggregationen benötigt und so ergänzen sich CMDB- und Architekturwerkzeug ideal.

Für die Planung von Ist- und Soll-Szenarien werden Informationen in aller Regel nicht in voller Tiefe und Ausführlichkeit benötigt. Hier bedeutet oft weniger mehr. Jedes Unternehmen muss den „Goldenen Schnitt“ finden, der soweit wie möglich abstrahierte Informationen

verwendet, aber auch die notwendige Detailierung enthält. Sammelt man zu viele Informationen, erhöht sich der Pflegeaufwand unnötig. Hat man zu wenig, kann man interessante Fragestellungen nicht mehr beantworten. Zu wissen, welche Fragen mit der Unternehmensarchitektur beantwortet werden sollen, ist der Schlüssel, hier die richtige Balance zu finden. Solche Fragestellungen könnten sein:

- Welche Anwendungen kommen im Bereich xyz zum Einsatz? (Objekte identifizieren)
- Was bedeutet die Änderung der Eigenschaft abc für mein Unternehmen? (Attribute definieren)
- Welche Abhängigkeit besteht zwischen meiner Infrastruktur xyz und meinen Prozessen abc? (geeignete Relationen (Verbindungen) aufsetzen)
- Welche Standards gibt es in meinem Umfeld bereits, auf die ich aufsetzen könnte? (bestmögliches Framework festlegen)

Rational System Architect ist flexibel und offen und kann einfach und schnell an unterschiedliche Unternehmensgegebenheiten angepasst und mit anderen Systemen integriert werden. Eine Reihe von Standards wird mitgeliefert, aus denen man auswählen kann, die sich teilweise kombinieren lassen und die ebenfalls jederzeit erweiterbar oder anpassbar sind. Frameworks bieten eine Standardisierung

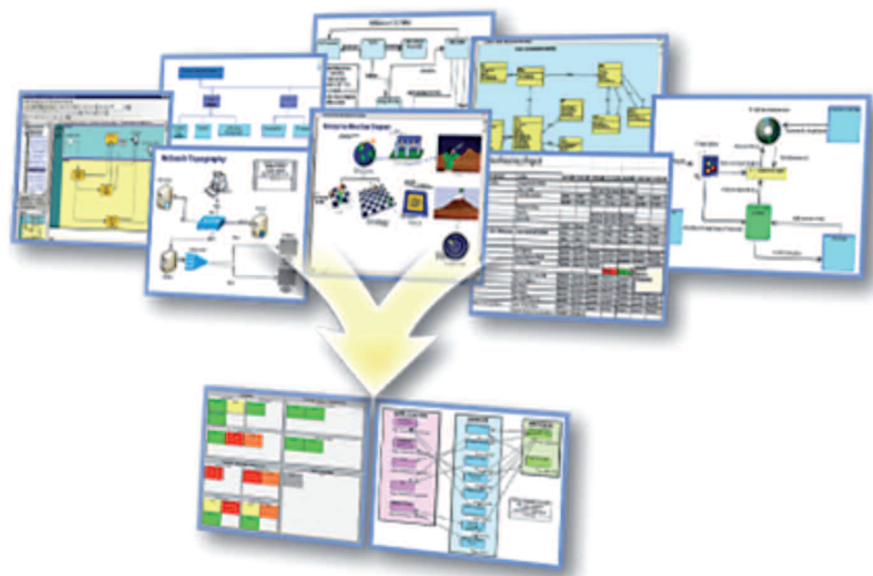


Abb. 2: Daten- und Informationskonsolidierung

und Normierung von Prozessen, Modellen oder auch Berichten an und erleichtern so den Einstieg in EA. Ähnlich einer Bebauungsplanung für eine Stadt, die entsprechende Rahmenparameter für die Architekturen in diesem Bereich festlegt, bieten sie einen Rahmen für die lagespezifischen Einflussfaktoren im Softwareumfeld. Sie helfen dabei, den Gesamtblick nicht zu verlieren, auch wenn aufgrund der aktuellen Tagesthemen erst einmal nur ein Teilbereich wie z. B. die Applikationslandschaft betrachtet wird. Und auch bei Veränderungen bilden sie einen stabilen Rahmen, in dem immer der Gesamtkontext zur Verfügung steht.

Dabei variieren die Umfänge der Frameworks sehr stark. Bietet TOGAF eher einen offenen Rahmen für viele verschiedene Branchen und geht von einer starken Adaption für das einzelne Unternehmen aus, so steht auf der anderen Seite ITIL mit genau vordefinierten Prozessen und Referenzdokumenten oder IFW mit vordefinierten Prozessen und Datenmodellen speziell für Banken zur Verfügung. Weitere Frameworks die von Rational System Architect unterstützt werden sind beispielsweise Zachman, NGOSS (Telekom), FEAF (Government), NAF, DoDAF, MoDAF (Militär). Jedes bietet für seinen Themenbereich die ideale Unterstützung, wird jedoch oftmals kombiniert, ergänzt oder abgespeckt für den entsprechenden Einsatzzweck im Unternehmen.

Anwendungsszenarien

Je weiter sich ein Framework mit Informationen füllt und eine Unternehmensarchitektur in ihrem Reifegrad vorwärts schreitet, umso vielfältiger werden die Informationen im System und umso größer wird die Anzahl der Benutzer der Plattform. Die Interessen und Notwendigkeiten der einzelnen Personen können dabei sehr unterschiedlich sein. Mögliche Rollen und Aufgabenstellungen der Nutzer müssen berücksichtigt und bedient werden:

- Architekten, die Modelle aus fachlicher Sicht zusammenstellen
- Administratoren, die Pflege- und Anpassungsaufgaben wahrnehmen
- (Detail-) Verantwortliche, die Informationen pflegen
- Anwender und Benutzer, die entsprechend aufbereitete Ergebnisse für ihre tägliche Arbeit benötigen

Weitere Rollen sind denkbar und die Liste ist so vielfältig erweiterbar wie die verschiedenen Unternehmensarchitekturen eben auch. Für die verschiedenen Bearbeiterrollen bieten sich unterschiedliche Schnittstellen an. Ein Architekt sollte vollen Zugriff auf alles haben, um den Gesamtkomplex gliedern zu können. Seine Benutzeroberfläche wird mehr Menüpunkte enthalten als die eines Detail-Verantwortlichen. Dieser benötigt zur Pflege seiner Anwendungsinformationen nur eingeschränkten Zugriff auf einzelne Objekttypen oder Diagramme. Ein Zugriff für ihn über einen Browser ist flexibel und reduziert zudem den Installationsaufwand für diesen Benutzerkreis. Der Endanwender arbeitet am Besten mit einer fertigen Publikation, die auf ihn zugeschnitten ist und einen definiert freigegebenen Stand der Information enthält. Sehr gute Ergebnisse werden hier mit Publikationen im Intranet gemacht, die sich durchgängig navigieren lassen und so eine gute Transparenz der Unternehmensarchitektur bieten. Die Möglichkeiten vom Rational System Architect sind hier sehr vielschichtig.

Neben den verschiedenen Arten des Zugriffs muss auch ein entsprechendes Berechtigungssystem aufgesetzt werden, da ja nicht alle Informationen in einem solch umfangreichen System für alle gleichermaßen offen sein sollen. Dabei sollte in der Startphase, in der noch viele Dinge entwickelt und angepasst werden, das System flexibel und offen sein. Mit zunehmender Anwenderzahl und steigender Komplexität muss das Berechtigungssystem die Arbeit kanalisieren helfen und die Sicherheit im System erhöhen. Nur so kann sichergestellt werden, dass die verschiedenen Einflussfaktoren und Inhalte richtig und aktuell sind.

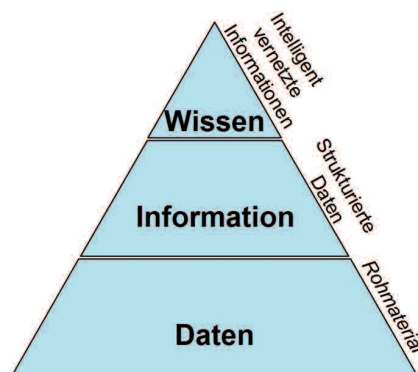


Abb. 3: Die Wissenspyramide

Transformation zum Wissen

Das Erfassen und Sammeln von Informationen in der Unternehmensarchitektur dient dazu, Entscheidungshilfen für die Verantwortlichen zur Verfügung zu stellen. Je mehr Analysen nun auf den gesammelten Daten möglich sind, umso größer ist der Nutzen, den die Unternehmensarchitektur leisten kann. Dabei zeigt uns die Wissenspyramide sehr schön, wie sich dieser Mehrwert aufbaut (siehe Abb. 3).

Gesammelt werden die verschiedenen Daten im Unternehmen. Durch die Strukturierung im System werden sie zu Informationen, die schon wesentlich nutzvoller sind. Aber erst durch die intelligente Vernetzung der Informationen wird aus dem Datenrohmaterial das Wissen um die Unternehmenslandschaft, das letztendlich für die Entscheidungsträger notwendig ist, um bessere Entscheidungen treffen, Prioritäten setzen, genauere Risikoabschätzungen machen und zukünftige Planungen effizienter gestalten zu können. Die Intelligenz der Vernetzung spiegelt sich in den Metastrukturen des Systems wider. Damit wird deutlich, wie wichtig die Flexibilität der unterstützenden Werkzeuge in Bezug auf das Metamodell zu bewerten ist. Rational System Architect zeigt hier seine Stärke und Offenheit, indem nicht nur Objekte und Attribute, sondern auch Beziehungen und Schnittstellen angepasst werden können.

Reporting

Die Auswertung des Wissens erfolgt vielfach über das Reporting. Da im EA Umfeld nicht nur einzelne Teilbereiche betrachtet und analysiert werden sollen, sondern die Informationsvielfalt in ihrem gesamten Spektrum, sind die Ansprüche und Vorstellungen in diesem Bereich ebenfalls sehr unterschiedlich und hoch. Flexible Formen für Dokumente und Online-Publikationen sind notwendig und sollten auch von den Anwendern entsprechend anpassbar sein. Eine mögliche Herangehensweise zur Nutzung der vernetzten Informationen soll das folgende Beispiel aus Rational System Architect darstellen.

Das Beispiel zeigt den Einfluss der Hardware auf die Geschäftsfunktionen. Dies könnte die Fragestellung „Was passiert, wenn ich den Server xyz abschalte?“ zur Grundlage haben. Die Verbindungslinien wurden über mehrere Ebenen zwischen Geschäftsfunktion und Hardware aufgesetzt:

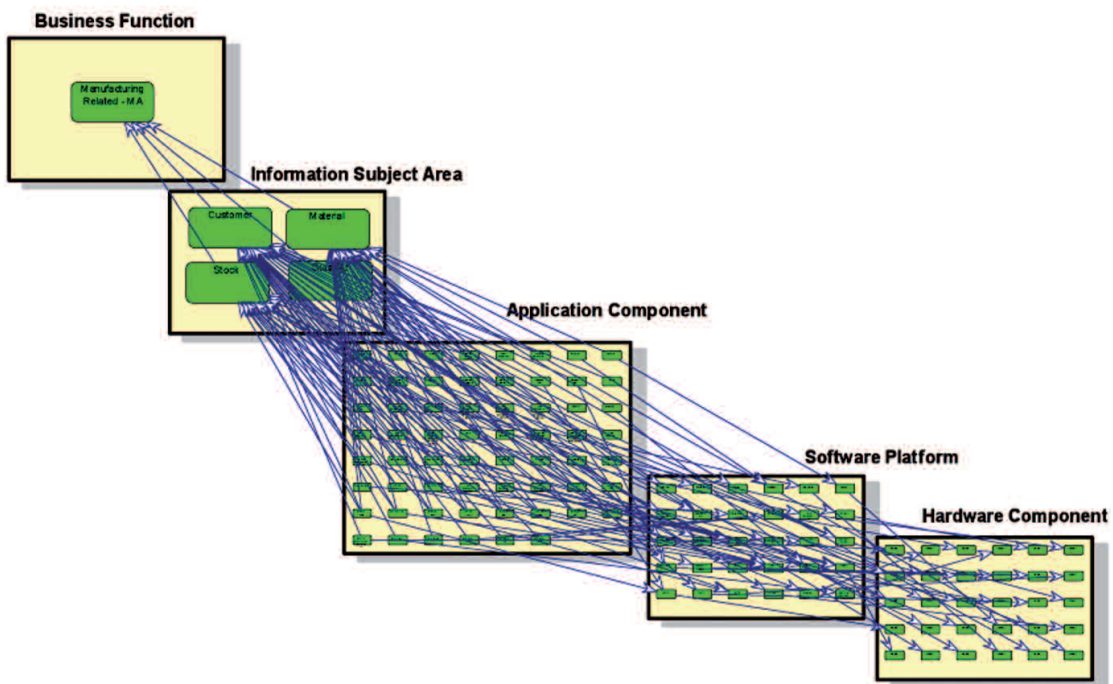


Abb. 4: Gesamtübersicht über alle Verbindungen

- Business Function
- Information Subject Area
- Application Component
- Software Platform
- Hardware Component

Die Auswertung erfolgt über eine Grafik in Form eines Explorer-Diagramms, in dem alle Informationsbereiche übergreifend ausgewertet werden. Über Explorer Object Reports wird eine entsprechende Teilmenge

von Daten ausgewählt und zur Anzeige gebracht. Mit Explorer Relationship Reports werden die Beziehungen zwischen den Objekten dargestellt. Das Ergebnis kann aufgebessert oder weiter detailliert werden. In **Abbildung 4** werden alle vorhandenen Beziehungen der Hardware-Komponenten über die verschiedenen Ebenen hinweg dargestellt.

Abbildung 5 zeigt einen spezifischen Pfad. So lässt sich ermitteln, was passiert,

wenn genau eine einzelne Hardware-Komponente ausfällt.

Weitere Analysen lassen sich über alle Daten- und Informationsebenen hinweg definieren. So können den verschiedensten Nutzergruppen passende Ergebnisse bereitgestellt werden. Es ist sinnvoll, Analysen möglichst klar zu kommunizieren. In diesem Beispiel kann das über einen Report in Papierformat geschehen, in dem mehrere Ebenen ausgeblendet werden. Dadurch ver-

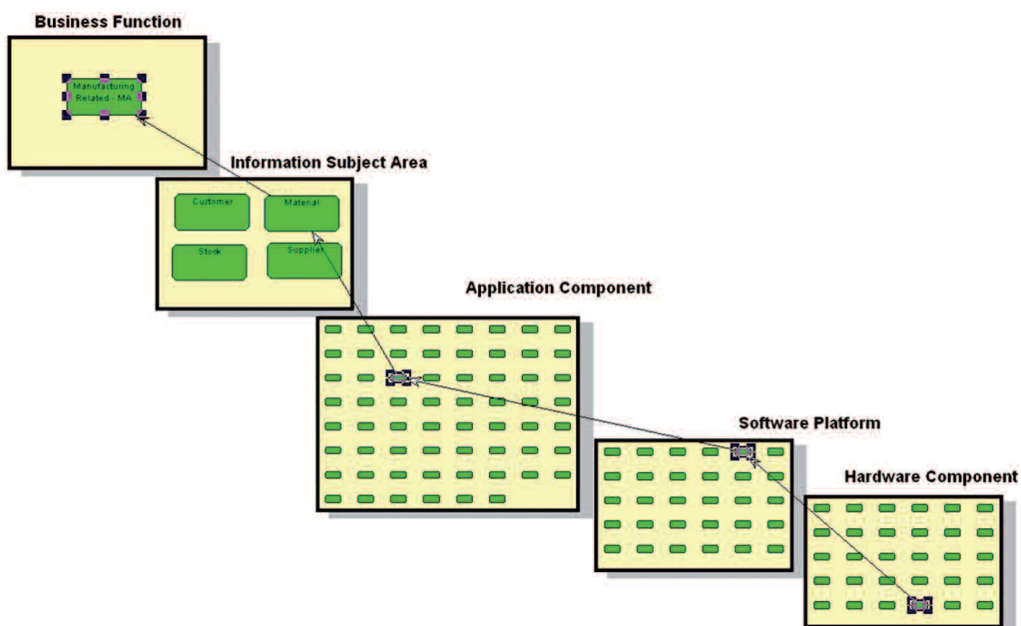


Abb. 5: Detail-Analyse einer Verbindungskette

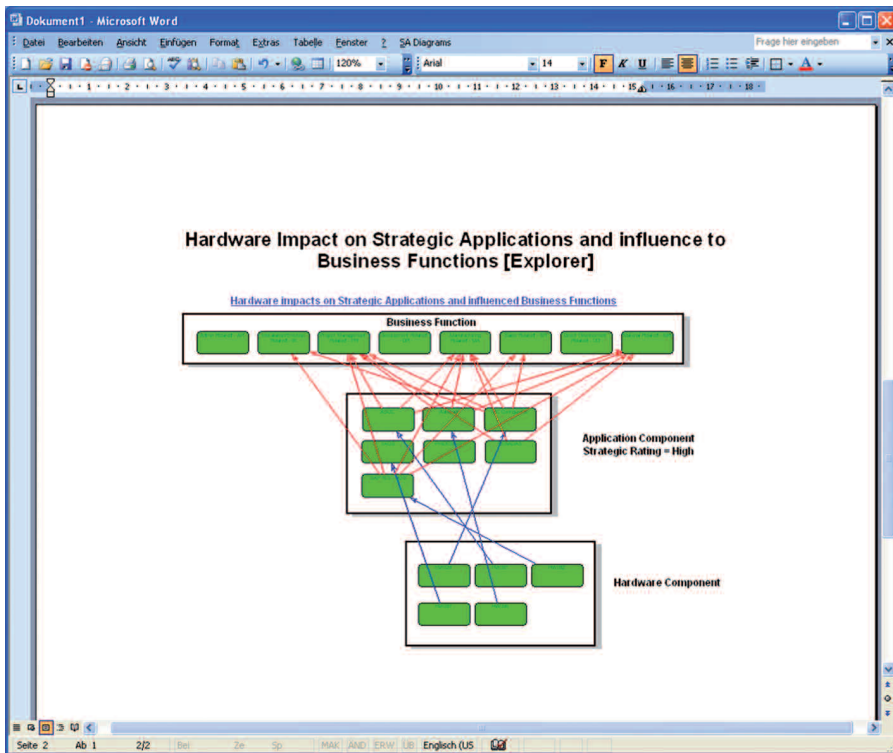


Abb. 6: Ein aufbereitetes Ergebnisdokument

einfach sich die Darstellung, der Informationsgehalt bleibt aber erhalten. **Abbildung 6** stellt eine so vereinfachte Analyse dar.

Fazit

Die im Beitrag besprochenen Situationen sind nur Beispiele für die vielfältigen

Möglichkeiten, wie Unternehmensarchitektur Information in Wissen verwandeln kann. Weitere Szenarien sind denkbar. Daher sollten die Analysemöglichkeiten, die Werkzeuge für Unternehmensarchitekturen unterstützen, flexibel sein. Um ihren Wert für die Wertschöpfungskette des Gesamtunternehmens darzustellen, muss

die Unternehmensarchitektur die Brücke zu den Geschäftsbereichen schlagen. Und wie eingangs im Vergleich zu Immobilien betrachtet, definiert sich hier die Lage eben auch im Gesamtkontext des Unternehmens - sowohl technisch als auch politisch und organisatorisch. Nur in dieser Gesamtsicht kann sie ihren vollen Wert darstellen. Am Ende ist sie zudem das beste interne Marketing-Mittel für den Wert der IT-Abteilung. Genauso wie sich Immobilien eben auch über eine gute Darstellung z. B. im Internet/Intranet besser verkaufen lassen, als wenn sie nur in einem Schaufenster des Immobilienbüros am Stadtrand hängen. ■

Quellen

[Sto02] Gernot Starke: Effektive Software-Architekturen, Hanser, 2002.
 [Rec00] Rehtin E., Maier M.: The Art of Systems Architecting, CRC Press, 2000.
 [Bom08] Christoph Bommer, Markus Spindler, Volker Barr: Softwarewartung, dpunkt.verlag, 2008.