

Managed Evolution: Nachhaltige Entwicklung großer Systeme

Christoph Pflügler

Proseminar IT-Kennzahlen und Softwaremetriken SS2010

Technische Universität München

Christoph.Pfluegler@mytum.de

Abstract: Die nachhaltige Weiterentwicklung von großen Anwendungslandschaften erweist sich als eine schwierige und komplexe Aufgabe. Strategische Fehler sind von großer Bedeutung für die betroffenen Unternehmen und sind oft nicht sofort ersichtlich. Deshalb ist eine Evolutionsstrategie bei der Weiterentwicklung von großen Anwendungslandschaften sinnvoll. Als eine mögliche Strategie stellt dieser Bericht das Konzept der Managed Evolution und das Vorgehen hierbei vor. Im zweiten Teil des Berichts wird detailliert auf die im Zusammenhang mit Managed Evolution verwendeten Metriken eingegangen. Hierbei wird einerseits eine Metrik zur Messung der IT-Entwicklungseffizienz, die aus drei verschiedenen Faktoren aufgebaut ist, sowie eine Metrik zur Bestimmung des Geschäftsnutzens präsentiert.

1 Notwendigkeit einer Evolutionsstrategie

„Als Anwendungslandschaften (*application landscape*) bezeichnen wir die Gesamtheit der Anwendungssysteme, die ein Unternehmen zur Organisation und Abwicklung seines Geschäfts betreibt“ [En08, S.65]. Sie sind zusammen mit den ihnen betreibenden Unternehmen über Jahre hinweg gewachsen und somit zu einem komplexen Gebilde geworden, dessen Weiterentwicklung sich aufgrund seiner Komplexität als schwierig erweist.

Ein Treiber der Komplexität ist Größe. Große Anwendungslandschaften bestehen aus tausenden von Anwendungssystemen und haben viele Millionen Zeilen von Code. Des Weiteren weisen die einzelnen Anwendungen eine hohe Integration untereinander auf und haben eine hohe Änderungsrate, da laufend neue Geschäftsanforderungen aus den Fachabteilungen umgesetzt werden. Ein weiterer Komplexitätstreiber ist der unterschiedliche technische Stand zwischen Teilen der Anwendungslandschaft. Der Grund hierfür ist die Tatsache, dass eine funktionstüchtige Anwendung über viele Jahrzehnte betrieben wird ehe sie ersetzt wird.

Die Weiterentwicklung ist nicht nur aufgrund der hohen Komplexität schwierig, sondern auch aufgrund der hohen Anforderungen, die von den Fachabteilungen an die Anwendungslandschaft und ihre Komponenten gestellt werden. Sicherheit, Verfügbarkeit und Zuverlässigkeit spielen hier eine große Rolle. Des Weiteren sollen sich neue Geschäftsanforderungen leicht und schnell umsetzen lassen um eine optimale Unterstützung der Geschäftsprozesse sicherzustellen. Es lastet auch ein großer Kostendruck auf der IT-Abteilung. Sie wird als Cost Center gemanagt und hat somit ein Budget, mit dem sie ihre beiden Hauptaufgaben, die Implementierung neuer Funktionen und den Betrieb der Anwendungslandschaft und seiner Komponenten, erfüllen muss.

Um diese Anforderungen vor dem Hintergrund der bestehenden Komplexität erfüllen zu können, wird eine Evolutionsstrategie für die Weiterentwicklung der Anwendungslandschaft und ihrer Komponenten benötigt. Im Folgenden soll die Managed Evolution, die eine mögliche Evolutionsstrategie darstellt, erläutert werden.

2 Managed Evolution

2.1 Das Konzept der Managed Evolution

Nach [Mu08, S.538] besteht die Managed Evolution aus folgenden drei Grundprinzipien:

- Investitionen in die Anwendungslandschaft werden nicht nur für die Umsetzung von Geschäftsnutzen verwendet, sondern auch für die Verbesserung der Architektur
- Die Evolution der Anwendungslandschaft besteht aus beherrschbaren und risiko-gesteuerten Evolutionsschritten. Zu große und nicht mehr kontrollierbare Evolutionsschritte bergen das Risiko des Scheiterns des Projekts
- Metriken, auf die im zweiten Teil dieses Berichts eingegangen wird, werden zur Kontrolle des Standes der Anwendungslandschaft verwendet

Der Geschäftsnutzen von Anwendungslandschaften besteht in der Unterstützung der Geschäftsprozesse eines Unternehmens. Dies führt zu einer Verbesserung des Geschäftsergebnisses, da die Prozesse effizienter und somit kostengünstiger ablaufen. Damit der Geschäftsnutzen realisiert werden kann, stellen die Fachabteilungen Anforderungen an die Anwendungssysteme. Diese sollen dann möglichst durch bereitgestellte Funktionen der Anwendungslandschaft erfüllt werden.

Ein wichtiger Aspekt bei der Implementierung von neuer Funktionalität ist die IT-Entwicklungseffizienz. Je höher sie ist, desto mehr Geschäftsanforderungen können bei identischem Aufwand umgesetzt werden. Die IT-Entwicklungseffizienz ist eine globale Eigenschaft und hängt von vielen Faktoren, wie den Fähigkeiten oder der Lohnstruktur der Entwickler, ab. Sie aber wird zu einem großen Teil durch die Architektur der Anwendungslandschaft beeinflusst. Deshalb wird die IT-Entwicklungseffizienz im Zusammenhang mit der Managed Evolution als eine Kennzahl für die Architektur der Anwendungslandschaft verwendet. Nach [Sc09, S. 65] ist Architektur als „*the fundamental organization of a system embodied in its components, their relationship to each other and to the environment, and the principles guiding its design and evolution*” definiert.

Um die Architektur zielbringend weiterzuentwickeln, ist die Beachtung wichtiger Architekturparadigmen notwendig. Im Folgenden werden ein paar dieser Paradigmen vorgestellt. Die Partitionierung besagt, dass große Systeme nach ihrer fachlichen Kohäsion aufgeteilt werden sollen. So wird sichergestellt, dass Teile der Anwendungslandschaft, die zusammen gehören, an derselben Stelle implementiert werden. Ein weiteres Paradigma ist die Komplexitätsreduktion, die unter anderem die Vermeidung und Eliminierung von Redundanzen zur Aufgabe hat. Eine Modularisierung und Kapselung von Teilen der Anwendungslandschaft verhindert unnötige Abhängigkeiten verhindern. Die Kopplung der Anwendungssysteme soll hierbei über bereitgestellte Services realisiert werden. Bei der Anwendungsentwicklung empfiehlt sich die Einführung von standardisierten Architekturpatterns um eine fachliche und architektonische Integrität sicherzustellen [Mu08, S. 544f].

Um das Konzept der Managed Evolution zu visualisieren, bietet sich ein Koordinatensystem an, bei dem die Abszisse den Geschäftsnutzen und die Ordinate die IT-Entwicklungseffizienz misst. Jeder Punkt im Koordinatensystem zeigt den Zustand der Anwendungslandschaft zu einem bestimmten Zeitpunkt. Die Evolution geschieht mit Hilfe von Projekten. Jedes Projekt liefert einen Beitrag zum Geschäftsnutzen und zur IT-Entwicklungseffizienz und sie werden durch Pfeile dargestellt. Die Summe aller Projekte, bzw. der Pfeile im Koordinatensystem, ergibt die Evolutionstrajektorie, also den Entwicklungspfad der Anwendungslandschaft. Die folgende Abbildung zeigt den Zustand einer Anwendungslandschaft zu sieben verschiedenen Zeitpunkten und die Realisierung von sechs Projekten.

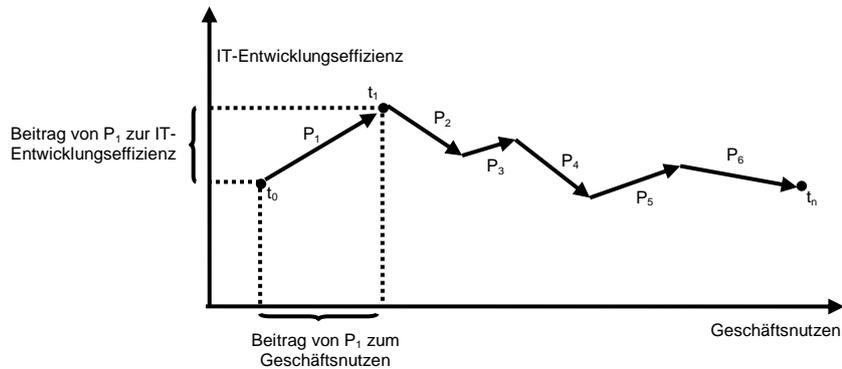


Abbildung 1: Koordinatensystem der Managed Evolution (nach [Mu08, S.540])

Man steht bei der Weiterentwicklung der Anwendungslandschaft vor der Frage, wie viel man in die Verbesserung der Architektur und wie viel in die Erfüllung von Geschäftsanforderungen investieren soll. Eine starke Konzentration auf die Architektur verbessert die IT-Entwicklungseffizienz, aber es wird zu wenig Geschäftsnutzen generiert. Die Anwendungslandschaft verursacht hohe Kosten, aber die Geschäftsprozesse werden nur unzulänglich unterstützt. Diese Form der Evolution existiert in der Praxis nicht, da eines der primären Ziele der IT die Erfüllung von Anforderungen aus den Fachabteilungen ist. Eine starke Konzentration auf die Erfüllung von Geschäftsanforderungen lässt die IT-Entwicklungseffizienz kontinuierlich sinken. Folgen sind beispielsweise Daten-Redundanzen, enge Kopplung und technologische Divergenz. Hierdurch erhöhen sich die EDV Kosten und die Implementierungsdauer verlängert sich. Des Weiteren kommt es zu einer „Verkrustung“ der Anwendungslandschaft. Sie wird so inflexibel, dass eine Evolution kaum mehr möglich ist. Oft ist ein Totalersatz des Systems die einzige Möglichkeit. Dieser dauert aber in der Regel mehrere Jahre und verursacht extrem hohe Kosten. Aus diesem Grund ist der Totalersatz als Evolutionsstrategie für große Systeme ungeeignet.

Die Lösung, die hierfür von der Managed Evolution vorgeschlagen wird, ist die Balance zwischen dem Geschäftsnutzen und der Architektur. Sie definiert einen Evolutionskanal, in dem die Evolutionstrajektorie verlaufen soll. Die Projekte müssen auch so gewählt werden, dass der Kanal nicht verlassen wird. Falls es doch dazu kommen sollte, werden spezielle Projekte durchgeführt, die die Evolutionstrajektorie wieder zurück führen. Der Kanal zeigt nach oben rechts und versucht die Anwendungslandschaft in Richtung des gewünschten Endzustandes zu entwickeln. Dieser repräsentiert eine Anwendungslandschaft mit hohem Geschäftsnutzen und hoher IT-Entwicklungseffizienz bzw. idealer Architektur.

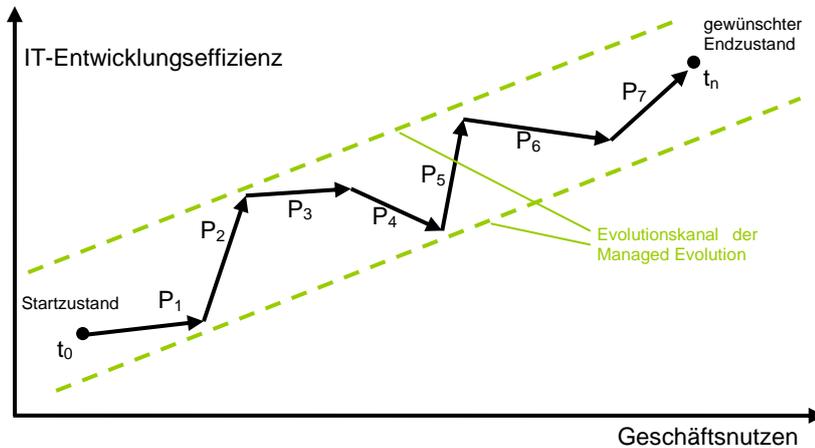


Abbildung 2: erweitertes Koordinatensystem der Managed Evolution (nach [Mu08, S. 541])

2.2 Das Vorgehen bei der Managed Evolution

Eine Voraussetzung für die Durchführung der Managed Evolution ist, dass die langfristigen und geschäftsgetriebenen Ziele der Weiterentwicklung der Anwendungslandschaft erhoben und bewertet worden. Des Weiteren muss die ideale Architektur der Anwendungslandschaft bekannt sein. Diese leitet sich von der Geschäftsarchitektur, die den Aufbau der Geschäftsprozesse bestimmt, ab.

Der erste Schritt ist die Erhebung der Ist-Anwendungslandschaft, die die existierenden Anwendungslandschafts-Komponente und deren Schnittstellen umfasst. Hierzu werden Interviews mit den Komponentenverantwortlichen durchgeführt und anschließend ausgewertet. Nachdem man ein Model der Ist-Anwendungslandschaft erstellt hat, wird diese operativ und architekturstrategisch bewertet um Handlungsbedarfe zu bestimmen. Bei der operativen Bewertung erfolgt eine Analyse der Abdeckung von Geschäftsanforderungen durch die bestehenden Komponenten der Anwendungslandschaft. Bei der architekturstrategischen Bewertung wird die Differenz zwischen der Ist-Anwendungslandschaft und der idealen Architektur der Anwendungslandschaft, die der Architektur des gewünschten Endzustandes entspricht, ermittelt. Auf Basis dieser beiden Bewertungen wird ein Hauptszenario bestimmt, das mehrere Maßnahmen bzw. Projekte zum Umbau der Anwendungslandschaft enthält. Hierbei werden die architekturstrategischen und geschäftsgetriebenen Handlungsbedarfe so ausgewählt und zu Projekten zusammengefasst, dass der Kanal der Managed Evolution nicht verlassen wird.

Als nächster Schritt wird die Soll-Anwendungslandschaft, die dem Zustand der Anwendungslandschaft nach Umsetzung der Hauptszenarien entspricht, modelliert. Zuletzt wird in der Roadmap der Weg von der Ist-Anwendungslandschaft zur Soll-Anwendungslandschaft qualitativ und quantitativ beschrieben. Die einzelnen Schritte der Umsetzung werden zeitlich bestimmt und der Aufwand der Maßnahmen wird geschätzt.

3 Metriken bei der Managed Evolution

Metriken spielen bei der Managed Evolution eine sehr große Rolle. Sie werden benötigt um den aktuellen Stand der Anwendungslandschaft zu messen und um die Evolution im Koordinatensystem der Managed Evolution verfolgen zu können. Man benötigt eine Metrik für die Bestimmung des Geschäftsnutzens und eine Metrik für die Messung der IT-Entwicklungseffizienz. Diese werden im Folgenden genauer erläutert.

3.1 Metrik für den Geschäftsnutzen

Der Geschäftsnutzen einer Anwendung besteht aus der von ihr gelieferten Funktionalität. Einerseits kann der Geschäftsnutzen aus der Steigerung des Umsatzes bestehen. Die Steigerung wird erst durch die Implementierung einer neuen Anwendung in die Anwendungslandschaft ermöglicht. Ein Beispiel hierfür wäre die Einführung eines Customer-Relationship-Management-Systems, das einen besseren Kundenservice für die bestehenden Produkte ermöglicht. Die Kundenzufriedenheit wird hierdurch gesteigert, was zu einem höheren Umsatz führt.

Andererseits kann der Geschäftsnutzen auch aus der Senkung von Kosten bestehen, die durch die Optimierung eines Prozesses durch eine Anwendung ermöglicht wird. Eine Möglichkeit zur Bestimmung der Kostensenkung, ist die Kalkulation der Menge von Arbeitskräfte-Tagen, die eingespart werden können. Hierbei wird zuerst bestimmt, um welche Zeitspanne sich der bestehende Prozess durch die neue Anwendung beschleunigen lässt und wie oft dieser Prozess pro Tag auftritt. Mit Hilfe der Kosten eines Arbeiters pro Tag kann berechnet werden, wie hoch die Einsparung ausfallen wird.

Da der Geschäftsnutzen einer neuen Anwendung erst in der Zukunft und über einen längeren Zeitraum auftreten wird, muss der Wert des zukünftigen Geschäftsnutzens zum jetzigen Zeitpunkt bestimmt werden. Hierfür bietet sich die Kapitalwertmethode an, die aus der betriebswirtschaftlichen Investitionsrechnung stammt und hier in einer vereinfachten Form dargestellt ist [Sp96, S. 432ff.]:

$$\text{Kapitalwert} = -I + \sum_{t=0}^T \frac{z_t}{(1+i)^t}$$

Der Kapitalwert einer Investition hängt im Wesentlichen von vier Faktoren ab. I ist die Investitionssumme des Projekts. Der Parameter i gibt die Höhe des Zinssatzes an und hängt von den Kosten der Bereitstellung der Investitionssumme ab. Ein Unternehmen muss sich für eine Investition entweder Kapital leihen, wofür es dann z.B. an eine Bank Zinsen zahlen muss, oder es kann die Investition aus eigenen Mitteln finanzieren. Hierbei fallen aber Opportunitätskosten an, da die Firma das Kapital auch an anderer Stelle investieren hätte können. Der Parameter z repräsentiert die Einsparungssumme, die durch die Realisierung der Investition ermöglicht wird. Der Faktor T bestimmt den Zeitraum in der Zukunft, in der die Einsparung auftritt. Die Kapitalwertmethode ermöglicht es, die Einsparungen „durch Diskontierung auf die Gegenwart vergleichbar zu machen“ [Hu09, S. 129]. Eine zentrale Aussage ist, dass je weiter eine Einsparung in der Zukunft liegt, desto weniger ist die Einsparung nach jetzigem Maße wert.

Im Allgemeinen sind nur Projekte mit positivem Kapitalwert erwünscht, aber manchmal werden auch Projekte mit negativem Kapitalwert durchgeführt. Eine neue gesetzliche Anforderung an die Anwendungslandschaft kann ein Projekt mit negativem Kapitalwert auslösen. Ein Projekt mit einem negativen Kapitalwert ist nicht immer schlecht, da dieses den Kapitalwert von anderen, nachfolgenden Projekten verbessern kann. Architekturprojekte, die in der Regel einen negativen Kapitalwert aufweisen, verbessern die IT-Entwicklungseffizienz und ermöglichen es, dass bei nachfolgenden Projekten mit gleicher Investitionssumme mehr Anforderungen aus den Fachabteilungen umgesetzt werden können, was zu einer höheren Einsparungssumme führt, bzw. dass zur Erreichung der selben Einsparungssumme eine geringere Investition nötig ist.

3.2 Metrik für die IT-Entwicklungseffizienz

Für die Bestimmung der IT-Entwicklungseffizienz einer Anwendungslandschaft benötigt man drei Messgrößen.

Die erste Messgröße ist die Lieferzeit (Time-to-Market) einer Anwendung. Sie ist definiert als die Zeitspanne zwischen dem Start des Projekts und der Übergabe der Anwendung. Die Bestimmung der Lieferzeit eines Projekts ist relativ einfach möglich, da sie ohne großen Aufwand und objektiv gemessen werden kann [Mu08, S.543].

Die zweite Messgröße, die benötigt wird, sind die Entwicklungskosten (Development Costs). Darunter versteht man alle Kosten, die von Start bis Abschluss des Projektes angefallen sind. Der Abschluss des Projekts ist nicht gleich der Übergabe der Anwendung, da auch noch eine Garantiefrist gewährt wird, in der in der Regel noch Fehler in der Anwendung auftreten, die behoben werden müssen. Diese Fehlerbehebung verursacht Kosten. Die Dauer einer Garantiefrist beträgt in der Regel ungefähr drei Monaten, kann aber normalerweise flexibel verhandelt werden [Mu08, S. 543].

Die dritte benötigte Messgröße ist eine Aufwandsschätzung des Projekts (Size). Prinzipiell kann jede Methode zur Bestimmung des Projektaufwands verwendet werden. Im Zusammenhang mit der Managed Evolution wird die Use-Case-Points (UCP) Methode verwendet. Sie geht davon aus, dass der Umfang eines Anwendungsentwicklungsprojektes maßgeblich durch die Anzahl und Komplexität der Interaktion mit der Anwendung und der Use-Cases der Anwendung beeinflusst wird. Die Use-Case-Points-Methode besteht aus zwei Gewichtungen. Die Aktorengewichtung teilt die Aktoren, die z.B. andere Anwendungen oder End-User darstellen, je nach der Komplexität ihrer Interaktion in drei Gruppen ein. Dabei werden jedem Aktor ein, zwei oder drei Punkte zugewiesen [Fr06, S. 3f.].

Aktorentyp	Aktoren Gewicht
Einfach	1 Punkt
Mittel	2 Punkte
Komplex	3 Punkte

Die Gewichtung der Use-Cases geht ähnlich vor. Die einzelnen Use-Cases werden nach ihrer Komplexität in Gruppen eingeteilt und ihnen werden entweder fünf, zehn oder fünfzehn Punkte zugewiesen [Fr06, S. 3f.].

Use-Case-Typ	Use-Case-Gewicht
Einfach	5 Punkte
Mittel	10 Punkte
Komplex	15 Punkte

Hier sieht man, dass die Use-Cases den Umfang eines Projekts mehr beeinflussen als die Aktoreninteraktion. Die Summe aller Aktorengewichte und aller Use-Case-Gewichte ergeben den unbereinigten Use-Case-Points. Den bereinigten Use-Case-Points erhält man durch die Multiplikation des unbereinigten Use-Case-Points mit zwei Faktoren [Fr06, S. 3f.]:

$$\text{bereinigte UCP} = \text{unbereinigte UCP} * \text{TCF} * \text{EF}$$

Der Technological Complexity Factor (TCF) gibt Aufschluss über technischen Randbedingungen des Entwicklungsprojekts. Zu diesen zählen unter anderem Performance, Zuverlässigkeit und Reusability. Sollen Teile der Anwendung bei einer anderen Entwicklung wiederverwendet werden, so steigert sich dadurch der Projektaufwand, da spezielle Anforderungen für eine Wiederverwendung erfüllt werden müssen. Ähnlich verhält es sich bei der Zuverlässigkeit. Soll die Anwendung eine sehr hohe Zuverlässigkeit aufweisen, steigert sich der Testaufwand. Der Wertebereich des Technological Complexity Factors liegt zwischen 0,6 und 1,3 [Fr06, S. 3f.].

Der Environmental Factor korrigiert die ungewichteten Use-Case-Points um den Einfluss der organisatorischen Komplexität und des Umfelds der Projektarbeit. Arbeiten in einem Projekt viele Mitarbeiter nur halbtags oder ist ihre Qualifikation eher niedrig steigert sich dadurch der Projektumfang. Die Werte des Environmental Factors liegen zwischen 0,425 und 1,7 [Fr06, S. 3f.].

Nachdem man die drei benötigten Messgrößen ermittelt hat, erfolgt eine Normierung der Lieferzeit und der Entwicklungskosten über den Projektaufwand, der den zuvor bestimmten Use-Case-Points des Projekts entspricht. Es wird also einerseits die benötigten Tage zur Realisierung eines Use-Case-Points sowie die Kosten pro Use-Case-Point berechnet [Mu08, S. 543]:

$$\tau = \frac{\text{Lieferzeit}}{\text{Projektaufwand}} \quad \sigma = \frac{\text{Entwicklungskosten}}{\text{Projektaufwand}}$$

Die IT-Entwicklungseffizienz ist definiert als der Kehrwert des Produkts dieser beiden Größen [Mu08, S. 543]:

$$\text{Entwicklungseffizienz} = \frac{1}{\tau * \sigma}$$

Hier sieht man, dass die Beschleunigung eines Projekts durch den Einsatz von mehr Mitarbeitern keinen Einfluss auf die Entwicklungseffizienz hat. Die Kosten pro Use-Case-Point steigen, aber gleichzeitig sinkt die Anzahl der benötigten Tage pro Use-Case-Point.

4 Ausblick und Bewertung der Managed Evolution

Die Managed Evolution ist eine wirkungsvolle Evolutionsstrategie für große Anwendungslandschaften. Leider zeigt sie aber nur Trends in Bezug auf die Architektur und die IT-Entwicklungseffizienz auf. Eine Investition in die Architektur steigert die IT-Entwicklungseffizienz. Die IT-Entwicklungseffizienz hängt aber nicht nur von der Architektur der Anwendungslandschaft, sondern auch von vielen anderen Faktoren, wie der Qualifikation oder den Lohnkosten der Mitarbeiter. Deshalb kann die Managed Evolution keine genau Aussage darüber treffen wie stark sich die IT-Entwicklungseffizienz bei einer Investition in die Architektur verbessert. Außerdem ist es allgemein sehr schwer zu bestimmen wie viel gute Architektur wert ist. Unternehmen fragen sich ob es finanziell besser ist weniger in die Architektur zu investieren und dafür eine niedrigere IT-Entwicklungseffizienz in Kauf zu nehmen. Viele CIOs wünschen sich eine Methode, die eine finanziell optimale Evolution der Anwendungslandschaft ermöglicht. Die Managed Evolution schafft eine Balance zwischen Investitionen in den Geschäftsnutzen und Investitionen in die Architektur. Ob diese Balance finanziell optimal ist, wird von ihr aber nicht aufgezeigt.

Literaturverzeichnis

- [En08] **Engels, G.; Hess, A.; Humm, B.; Juwig, O.; Lohmann, M.; Richter, J.; Voß, M.; Willkomm, J. (2008):** Quasar Entreprise – Anwendungslandschaften serviceorientiert gestalten, 1. Auflage, dpunkt Verlag, Heidelberg
- [Fr06] **Fronhoff, S.; Jung, V.; Engels, G. (2006):** Use Case Points in der industriellen Praxis, In: http://www.ch.capgemini-sdm.com/web4archiv/objects/download/pdf/1/sdm_pub_metrikon_fronhoff.pdf,
zugegriffen am: 08.07.2010
- [Hu09] **Hutzschenreuter, T. (2009):** Allgemeine Betriebswirtschaftslehre: Grundlage mit zahlreichen Praxisbeispielen, 3. Auflage, Gabler Verlag, Wiesbaden
- [Mu08] **Murer, S.; Worms, C.; Furrer, F. (2008):** Managed Evolution – Nachhaltige Weiterentwicklung großer Systeme, In: Informatik-Spektrum, Vol. 39, Nr. 6, S.537-547
- [Sc09] **Schmidt, C. (2009):** Management komplexer IT-Architekturen: empirische Analyse am Beispiel der internationalen Finanzindustrie, 1. Auflage, Gabler Verlag
- [Sp96] **Spremann, K. (1996):** Wirtschaft, Investition und Finanzierung, 5. Auflage, Oldenbourg