

# Projektübergreifendes Management – Der strategische Applikationslebenszyklus am Beispiel des BMW Q-Cockpit

Andreas Györy · Günter Seeser · Anne Cleven · Falk Uebernickel · Walter Brenner

Eingegangen: 14. Mai 2014 / Angenommen: 14. Juli 2014 / Online publiziert: 29. August 2014  
© Springer Fachmedien Wiesbaden 2014

**Zusammenfassung** Oft scheitern oder verzögern sich Softwareprojekte, weil die Veränderungsfähigkeit des Unternehmens überschätzt wird und zu viele Zielsetzungen gleichzeitig verfolgt werden. Basierend auf den Erkenntnissen aus einem Entwicklungsprojekt bei der BMW Group werden die Ziele für eine nachhaltige Softwareentwicklung und -einführung anhand eines strategischen Applikations-Lebenszyklus (SALZ) auf vier Evolutionsphasen verteilt. Diese verzahnte Entwicklung von Software und Geschäftsprozessen ist der Schlüssel für ein nachhaltiges und erfolgreiches Business-IT-Alignment (BITA) auf der Applikationsebene.

**Schlüsselwörter** Applikationsmanagement · Applikationsportfoliomanagement · Applikationslebenszyklus · Projektportfoliomanagement · Strategische IT Planung

**Abstract** When organizations misjudge their own ability to adapt during software development or rollout, application projects often fail to achieve their goals or stay within their planned time and budget. Based on the insights from an application implementation project at the BMW Group, the multiple goals of such an endeavor are structured into four strategic application lifecycle (SALC) stages. Establishing

---

A. Györy (✉) · F. Uebernickel · W. Brenner  
Institut für Wirtschaftsinformatik  
Universität St. Gallen  
St. Gallen, Schweiz  
E-Mail: andreas.gyoery@unisg.ch

G. Seeser  
Q-Monitoring und Q-Lenkung  
BMW Group,  
München, Deutschland

A. Cleven  
A<sup>2</sup> Research,  
Poschiavo, Schweiz

these stages allows an integrated software development and business process improvement and is one of the key factors for a successful Business IT Alignment (BITA) on the application level.

## 1 Herausforderung multipler Zielsetzungen

Die wesentlichen Zielsetzungen für unternehmensinterne Softwareprojekte (inkl. Entwicklung, Einführung, Erneuerung, Migration, Konsolidierung und Abschaltung) lassen sich auf vier übergeordnete Kategorien reduzieren:

- *Mehrwert für die Anwender bieten*  
Themenfeld: Anwenderorientierung (Hallerstede et al. 2011)
- *Strategischen Mehrwert für das Unternehmen bieten*  
Themenfeld: Business Engineering (Österle und Blessing 2005)
- *Effizienz operativer Prozesse steigern*  
Themenfeld: Process Performance Management (Cleven 2011)
- *Effizienz im Betrieb der Softwarelösung steigern*  
Themenfeld: IT-Kosten (Scheeg und Pilgram 2003)

Nachhaltig erfolgreiche Applikationen erfüllen im Laufe ihres Einsatzes Ziele in jeder dieser vier Kategorien. Softwareprojekte die jedoch Ziele und Anforderungen mehrerer Kategorien gleichzeitig verfolgen, übersteigen häufig die Veränderungsfähigkeit des Unternehmens und erfordern anschließend eine Kompensierung durch aufwändige Schulungen. So gibt es viele Beispiele gescheiterter ERP Einführungsprojekte bei denen durch eine Systemeinführung (Ziel der Akzeptanz durch Nutzer) gleichzeitig neue Geschäftsabläufe (Ziel der Prozessoptimierung) implementiert werden sollten. Solche ambitionierten Softwareprojekte, bei denen der Aufwand für die Veränderungen des Unternehmens (z. B. Anlernen eines neuen Systems und gleichzeitig neuer Geschäftsabläufe) unterschätzt wird, erleben häufig stark verzögerte Einführungen, drastische Personalwechsel oder frühzeitige Abbrüche (Ewusi-Mensah 2003). Zusätzlich erschweren die widersprüchlichen Anforderungen unterschiedlicher Kategorien die Entwicklung eines kohärenten Systemmodells. Deshalb betrachtet dieser Artikel die folgende Frage:

Wie kann die Entwicklung und Einführung einer Applikation in unabhängige Teilprojekte gegliedert werden, ohne dabei die Veränderungsfähigkeit des Unternehmens zu übersteigen und dennoch alle Zielsetzungen nachweisbar erfüllen?

Um diese Frage zu beantworten, wurde die Entwicklung des BMW Q-Cockpits untersucht und daraus ein strategischer Applikations-Lebenszyklus (SALZ) abgeleitet. Das Modell erlaubt es, einen Entwicklungs- und Einführungsprozess anhand von vier Evolutionsphasen in Teilprojekte zu gliedern. Im Mittelpunkt steht dabei die Zusammenfassung von Zielen und Anforderungen anhand der oben genannten Kategorien und deren Zuordnung zu den Evolutionsphasen. Dadurch ergibt sich ein Meta-

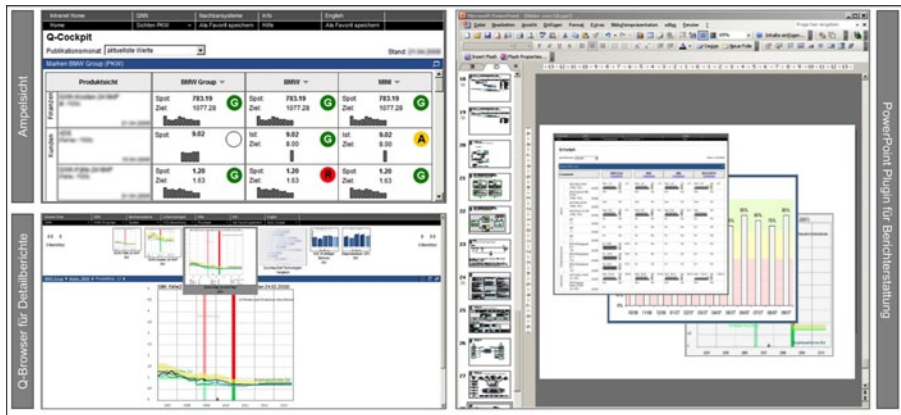


Abb. 1 Die drei Kernfunktionen des Q-Cockpit (in LS1)

modell, welches die Auswahl der Teilprojekte und die jeweiligen Entwicklungs- und Anforderungsmanagementmethoden sowie die Ressourcenverteilung unterstützt.

In Ergänzung zu gängigen Requirements Engineering und Management Methoden (siehe (Rupp et al. 2009)) bietet das Modell die Möglichkeit der frühzeitigen Identifikation und Korrektur (oder Terminierung) erfolgsgefährdeter Projekte durch messbare Teilziele und liefert einen übergeordneten Rahmen für die Anforderungspriorisierung.

Für die Analyse des Q-Cockpits wurden für den Betrachtungszeitraum von 2008 bis 2011 Anwenderumfragen, Nutzerzahlen und Zugriffsprotokolle berücksichtigt sowie das kumulative Fachkonzept<sup>1</sup> ausgewertet. Der technische und fachliche Projektleiter des Q-Cockpits beteiligten sich an der Beschreibung des Entwicklungs- und Einführungsprozesses für diesen Artikel.

## 2 BMW Q-Cockpit

Das Q-Cockpit (siehe Screenshots in Abb. 1) wurde im Rahmen einer unternehmensweiten Qualitätssteigerungsinitiative der BMW Group entwickelt und im September 2008 in der ersten Version zur Nutzung freigegeben. Das initiale Projekt umfasste zehn Releases und wurde 2011 um ein Folgeprojekt erweitert. Zum Zeitpunkt des Artikels wird der weitere Ausbau des Q-Cockpits bis 2016 projektiert.

Das strategische Ziel der Applikation ist es, Qualitätsdaten transparenter und schneller zugänglich zu machen, um mit einer gesteigerten Reaktionsfähigkeit im Qualitätsmanagement langfristig die Produkt- und Servicequalität der Unternehmensgruppe zu verbessern. Zusammen mit den Ergebnissen flankierender Projekte unterstützt das Q-Cockpit heute die Früherkennung von Problemen, die Durchgängigkeit der Qualitätssteuerung, die Aufwandsminimierung im Qualitätsberichtsessen sowie die Vergleichbarkeit, Konsistenz und Validität von Qualitätsdaten.

<sup>1</sup> Stand Juni 2013.

Dafür greift das Q-Cockpit auf die Qualitätsmanagement-Datenbasis zu. Diese wurde in einem begleitenden Projekt entwickelt und ist ein Zusammenschluss von Data Warehouses (DW, ein DW pro Ressort), um einen Single-Point-Of-Truth (SPOT) für alle Qualitätsdaten zu bilden. In weiteren Projekten wurden alle für das Qualitätsmanagement relevanten Kenngrößen und Berichtsprozesse vereinheitlicht. Die Ergebnisse dieser Projekte wurden im Q-Cockpit zusammengeführt, um eine zentrale Plattform zur Visualisierung, Überwachung, Kommunikation und Ablage aller Qualitätsinformationen zu bieten.

Die Auszeichnung mit dem Best Practice Award für Business Intelligence und Data Warehouse Lösungen des Business Application Research Center (BARC) (Bayer 2010) bestätigte 2010 die technische Überlegenheit gegenüber anderen Business Intelligence (BI) Lösungen. Die seit der Einführung steigenden Nutzerzahlen sind wiederum ein Indikator für den Mehrwert, den die Applikation ihren Anwendern bietet. Heute hat das Q-Cockpit mehr als 1200<sup>2</sup> täglich aktive Anwender aus allen Bereichen der BMW Group und von vielen Lieferanten.

Vor der Einführung des Q-Cockpits war das Qualitätsberichtswesen, basierend auf der Verteilung von PowerPoint- und Excel-Dateien per Email, aufwendig und sehr heterogen. Es wurden unterschiedliche Kenngrößen und inkonsistente Daten verwendet und viele Berichte wurden manuell erstellt und dezentral gespeichert. Die BMW Group hat es mit ihren Bestrebungen in weniger als zwei Jahren geschafft, die Berichtsprozesse über Unternehmens- und Landesgrenzen hinweg zu vereinheitlichen, die Qualitätsdaten zu zentralisieren und diese über eine einheitliche Plattform, das Q-Cockpit, miteinander zu verknüpfen.

### 3 Entwicklung des Q-Cockpits in drei Leistungsstufen

Das BMW Q-Cockpit wurde in drei inkrementellen Teilprojekten, sogenannten Leistungsstufen (LS), umgesetzt, die eine Identifikation der unterschiedlichen Merkmale der evolutionären Entwicklungsschritte der Applikation ermöglicht.

Tabelle 1 fasst die wichtigsten Merkmale zusammen, welche die einzelnen LS voneinander unterscheiden. Die Übersicht verdeutlicht wie die Aufteilung in LS es ermöglichte, die Funktionen, Inhalte und Nutzerkreise gestaffelt aufzubauen und gleichzeitig die Entwicklungsmethode und die eingesetzten Technologien an neue Erkenntnisse und Anforderungen anzupassen. Wie mit dem Q-Cockpit in den einzelnen LS Veränderungen der Geschäftsprozesse erreicht wurden, wird in den folgenden Kapiteln dargelegt.

#### 3.1 Leistungsstufe 1 (LS1) – „Quick-Win“

##### 3.1.1 Zielsetzung

Das Ziel war die Definition der technischen Architektur und des Rumpfes der Applikation für die Funktionalitäten und für die grafische Benutzeroberfläche. Dabei sollte das System von Anwendern möglichst schnell zielführend eingesetzt und evaluiert werden können.

<sup>2</sup>Stand Q1 2013.

**Tab. 1** Übersicht der Q-Cockpit Leistungsstufen des initialen Entwicklungsprojekts

Kriterien	Leistungsstufe 1	Leistungsstufe 2	Leistungsstufe 3
Fokus:	Anwendernutzen	Unternehmensziel	Optimierung
Konkrete Zielsetzung:	„Quick Win“ Funktionen	Hauptnutzen (Kenn- zahlen + Funktionen)	Nutzung erweitern + verbessern
Entwicklungszeitraum:	06.2007–09.2008 (15 Monate)	06.2008–12.2009 (18 Monate)	09.2009–02.2011 (17 Monate)
Nutzungszeitraum:	09.2008–05.2009 (6 Monate)	05.2009–05.2010 (12 Monate)	05.2010–08.2011 (15 Monate)
Dominierende Entwicklungsmethode:	Wasserfall	Anwenderintegriert, iterativ	Anwenderintegriert, iterativ
Interne Wertschöpfungstiefe:	Hoch	Hoch	Hoch/Reduziert <sup>a</sup>
Releases:	1	3	6
Vorgesehener Nutzerkreis:	Zentraler Qua- litätsbereich, Produktlinien	Zentraler Qualitäts- bereich, Produktlinien, Entwicklung, Produk- tion, Service	Alle Qualitäts- bereiche der BMW Group, Lieferanten
Aktive Anwender <sup>b</sup> :	ca. 400 <sup>c</sup>	528	820
Neue Kernfunktionen:	Ampelansicht, Q-Browser, PPT-Plugin	Alerting, Berichts- management, Expertenmodus	Onlineberichte
Neue Zusatzfunktionen <sup>c</sup> :	Keine	Kommentierung, Berechtigungsmanage- ment	Personalisierung, Kenngrößen-Cluster, Schnellnavigation
Neue Kenngrößen:	9	15	9
Neue Kenngrößen-dimensionen <sup>d</sup> :	3	4	4
Technologieänderungen <sup>e</sup> :	Initialarchitektur	IBM Cognos BI → Eigenentwicklung	Internetexplorer 7 → ChromeFrame

<sup>a</sup>Erstes Release/Fortlaufende Entwicklung

<sup>b</sup>Durchschnittliche Anzahl individueller Anwender die täglich zugreifen

<sup>c</sup>Da in der LS1 kein Monitoring durchgeführt wurde, liegen hier nur Schätzungen vor

<sup>d</sup>Berücksichtigt werden auch Erweiterungen einer bestehenden Dimension um weitere Ausprägungen

<sup>e</sup>Im Vergleich zur vorherigen LS

**Tab. 2** Technische Architektur der LS1

	Module	Technologie
Q-Cockpit	PowerPoint Plugin:	Microsoft.NET Framework 3.5
	Client:	InternetExplorer 7
	Frontend:	XHTML, JavaScript, JQuery
	Aggregation & Visualisierung:	IBM Cognos 8 BI
	Middleware:	Java J2EE
Q-Datenbasis	Datawarehouse <sup>a</sup> :	Oracle 10g
	Datenintegration (ETL):	Informatica 8.6.0

<sup>a</sup>Für Ressort-übergreifende Qualitätsdaten

### 3.1.2 Umsetzung

Gemäß den Unternehmensvorgaben für Softwareentwicklung und -architektur wurde das Q-Cockpit mit einer sequentiellen Entwicklungsmethode (Wasserfallmodell) und verfügbarer Standardsoftware (z. B. IBM Cognos; siehe auch Tab. 2) in einem Release implementiert. Um eine möglichst einfach und schnell einsetzbare Version

**Tab. 3** Unterstützung und Automatisierung (grau markiert) von Teilprozessen durch Office Applikationen und den Leistungsstufen des Q-Cockpits

Teilprozesse (grobe Abfolge):	Abruf	Aggregation	Übersicht	Visualisierung	Kommentierung	Überwachung	Erzeugung	Präsentation	Speicherung	Verteilung
	Unterstützende Applikationen:		Kenngroßen			Berichte				
Quellsysteme:	●									
Excel:		●	●	●	●					
PowerPoint:					●		●	●		
Dezentrale Datenablage:									●	
Outlook/ Email:										●
Q-Cockpit LS1:	●	●	●	●						
Q-Cockpit LS2:	●	●	●	●	●	●			●	●
Q-Cockpit LS3:	●	●	●	●	●	●	●	●	●	●

der Applikation bereitzustellen, wurde der angesprochene Anwenderkreis limitiert. Die bereitgestellten Funktionen und Inhalte wurden auf „Quick Wins“ beschränkt. „Quick Wins“ sind die Umfänge einer Applikation die bei verhältnismäßig geringem Entwicklungsaufwand den höchsten Mehrwert für Anwender bieten.

### 3.1.3 Evolution der Applikation

Das Q-Cockpit wurde als browserbasierte Intranet-Applikation zur freiwilligen Nutzung eingeführt. Es enthielt drei Kernfunktionen, welche die ersten vier Teilprozesse des Qualitätsmanagements (siehe hierzu auch Tab. 3) unterstützen:

Die *Ampelsicht* bietet Anwendern eine schnelle Übersicht über den Status aller für sie relevanten Qualitätswerte. Mit dem *Q-Browser* können begrenzte Recherchen anhand von Detailberichten unmittelbar im Q-Cockpit durchgeführt werden. Für ausführlichere Nachforschungen ist es möglich, über kontextbezogene Absprünge die jeweiligen Quellsysteme aufzurufen. Das *PowerPoint-Plugin* bietet die Möglichkeit, die Darstellungen der Ampelsicht und des Q-Browsers in eine lokale Präsentation (z. B. für Qualitätsberichte) als Grafik auszuleiten und dort auf Knopfdruck zu aktualisieren.

### 3.1.4 Evolution des Geschäftsprozesses

Die Nutzung des Q-Cockpits bedeutete für Berichterstatter und Führungskräfte die Reduktion des Erstellungsaufwandes für den monatlichen Qualitätsbericht auf einen Bruchteil. Mit der intuitiven Bedienbarkeit, der niedrigen Einstiegshürde und der

**Tab. 4** Applikationsklassen und SALZ-Phasen der Leistungsstufen

	Leistungsstufe 1	Leistungsstufe 2	Leistungsstufe 3	
Entwicklungsumfang:	„Quick-Win“-Funktionen	Hauptfunktionen	Zusatzfunktionen	kontinuierliche Weiterentwicklung
Fokus:	Anwendernutzen	Unternehmensziel	Prozessoptimierung	Ressourcen optimieren
Veränderung für Anwender	Neue Applikation	Neue Prozesse	Optimierte Applikation	
Strategisches Potential <sup>a</sup> :	Hoch	Hoch	Mittel	Niedrig
Operative Kritikalität <sup>a</sup> :	Keine	Hoch	Hoch	Hoch
Applikationsklasse:	High Potential	Strategic	Key Operational	Support
SALZ:	Anwenderphase	Strategiephase	Effizienzphase	Ressourcenphase

<sup>a</sup>Stand Juni 2013

konkreten Beschleunigung gängiger Arbeitsschritte etablierte sich daher der Einsatz der Applikation unter den Mitarbeitern in kurzer Zeit.

Über den direkten Anwendernutzen der Applikation hinaus wurden mit dem Erfolg der LS1 auch das Veränderungspotential im Qualitätsmanagement und die Eignung des Q-Cockpits dieses Potential zu heben, bestätigt.

### 3.2 Leistungsstufe 2 (LS2) – „Hauptnutzen erfüllen“

#### 3.2.1 Zielsetzung

Nachdem mit der LS1 das Q-Cockpit im Unternehmen für bestehende Arbeitsschritte etabliert wurde, war es das Ziel der LS2, das Qualitätsmanagement grundlegend zu verändern, um die Reaktionsfähigkeit auf Qualitätsprobleme zu verbessern.

Zusätzlich sollten auch der Anwenderkreis um die Qualitätsstellen verschiedener Ressorts erweitert und entsprechende Kennzahlen und Dimensionen aufgenommen werden.

#### 3.2.2 Umsetzung

Basierend auf den Erfahrungen aus der LS1 wurde die sequentielle Entwicklungsmethode durch ein iteratives anwenderintegriertes Vorgehen mit drei Releases abgelöst. Dies wurde von einem dedizierten Anforderungsmanagement flankiert, welches heute noch die Anmerkungen der Anwender transparent (über eine Intranet-Seite verfolgbar) in die Entwicklung einfließen lässt. Auch die Applikationsarchitektur wurde angepasst, um die wachsende Komplexität der Inhalte effizient abbilden und darstellen zu können. Die damalige Version der Standardsoftware IBM Cognos, deren Berichte für das Q-Cockpit manuell definiert und release-abhängig aktualisiert werden mussten, wurde durch eine Eigenentwicklung basierend auf Java (J2EE) und MonarchCharts ersetzt, die eine automatisierte Darstellung erlaubte.

### 3.2.3 Evolution der Applikation

Mit der LS2 wurde das Q-Cockpit um drei Kern- und zwei Zusatzfunktionen erweitert, mit denen zusätzliche Teilprozesse unterstützt und Prozessveränderungen ermöglicht wurden (siehe hierzu auch Tab. 3).

Das *Alerting-System* informiert Anwender per Email sofort über das Überschreiten eines Kenngrößenschwellwerts. Damit ist im Qualitätsmanagement zusätzlich zum zyklischen Pull-Prinzip von Berichten ein proaktiver Push von Informationen möglich, der die Reaktionsgeschwindigkeit um bis zu einen Berichtszyklus (in der Regel ein Monat) verbessert.

Das *Berichtsmanagement* ersetzte die manuelle, dezentrale Ablage und Verteilung der Qualitätsberichte (per Email) durch einen zentral gesteuerten, gesicherten und automatisierten Prozess. Diese Funktion wurde im Laufe der LS2 verbindlich im Entwicklungsressort eingeführt, um damit einen SPOT<sup>3</sup> auch für Qualitätsberichte<sup>4</sup> zu etablieren.

Der *Expertenmodus* bietet die Möglichkeit, Ansichten und Filtereinstellungen beliebig selbst zu konfigurieren, ohne dabei an die vordefinierten Ampelansichten gebunden zu sein. Dadurch kann das Qualitätsmanagement mit neuartigen Auswertungen bereichert werden.

Die neuen Kernfunktionen und die Erweiterung der Inhalte machte zwei Zusatzfunktionen notwendig: Kommentierung und Berechtigungen.

Die *Kommentierung* ermöglicht es, Inhalte im Q-Cockpit mit Erläuterungen zu versehen, die insbesondere für Kennwerte, die Alerts auslösen, von Bedeutung sind, da damit alert-bedingte Eskalation und Klärungsrunden vermieden werden können.

Mit dem *Berechtigungsmanagement* wurde zusätzlich zur bestehenden Anwenderauthentifizierung der Zugriff auf Daten im Q-Cockpit inhaltsbezogen eingeschränkt. Dies wurde besonders mit der steigenden Durchgängigkeit<sup>5</sup> des Kenngrößensystems und der wachsenden Anwenderanzahl notwendig, um die Richtlinie für die Datenbefugnis unterschiedlicher Rollen im Unternehmen zu erfüllen.

### 3.2.4 Evolution des Geschäftsprozesses

Mit der LS2 wurde die niedrige Einstiegshürde der LS1 für neue Anwender erhalten, während gleichzeitig erfahrene Nutzer mit neuen Funktionen ihre Arbeitsschritte neu gestalten konnten. Diese anwendergetriebene Evolution der Geschäftsprozesse bewirkte die nachhaltige Verbesserung des Qualitätsmanagement im Rahmen der Veränderungsfähigkeit des Unternehmens. Diese Entwicklung wurde durch die unternehmensweite Verbreitung und Nutzung der Applikation nachhaltig gefördert.

Im Laufe der LS2 wurden unternehmensweite Prozessveränderungen zunehmend deutlich.

So wurde beispielsweise der Aufwand für die halbjährlichen Qualitätsberichte von ca. 200 Personentage auf zwei Personentage reduziert. Weiterhin konnte ein Großteil

<sup>3</sup> Single-Point-Of-Truth (siehe auch Kap. 2).

<sup>4</sup> Nur zentral gesteuerte Standard-Qualitätsberichte.

<sup>5</sup> Sowohl unternehmensweite und strategische als auch modulbasierte und operative Kenngrößen.



der qualitätsbezogenen Klärungsrunden zur Korrektheit und Stimmigkeit von Qualitätsdaten vermieden werden.

Mit der LS2 wurde, auf der etablierten Funktionsweise der LS1 aufbauend, das volle strategische Potential der Applikation realisiert und damit eine anwendergetriebenen Verbesserung angestoßen, die bis heute anhält.

### 3.3 Leistungsstufe 3 (LS3) – „Nutzung optimieren“

#### 3.3.1 Zielsetzung

Für die LS3 stand die Verbesserung der in LS1 und LS2 eingeführten Kernfunktionen im Mittelpunkt, um deren effiziente Nutzung zu fördern. Zusätzlich sollten die Inhalte und der Anwenderkreis erweitert und das Projekt in eine langfristige Weiterentwicklung überführt werden.

#### 3.3.2 Umsetzung

Da sich die Anwenderorientierung bewährt hatte, wurde die Entwicklungsmethode der LS2 fortgesetzt. Dabei wurde die Release-Häufigkeit erhöht, um die Reaktionsfähigkeit auf Anforderungen zu verbessern. Anwender-seitig wurde der Browser (Unternehmensstandard: InternetExplorer 7), durch ein Plugin (Google Chrome Frame) ergänzt, um damit die Darstellung des Cockpits auf ein von Anwendern akzeptiertes Niveau zu beschleunigen. Nach der Einführung der LS3 wurde mit der kontinuierlichen Weiterentwicklung des Q-Cockpits (weitere fünf Releases) ein externer Entwicklungspartner beauftragt, der das Projekt bereits ab der LS1 zunehmend unterstützte.

#### 3.3.3 Evolution der Applikation

Mit der LS3 wurden drei Zusatzfunktionen Personalisierung, Kenngrößen-Cluster und Schnellnavigation eingeführt, um die Nutzung der Kernfunktionen zu vereinfachen, während eine neue Kernfunktion „Onlineberichte“ die Lücke in der Berichtsprozessunterstützung schloss (siehe hierzu auch Tab. 3).

Mit den *Onlineberichten* können Abfolgen von Q-Cockpit Darstellungen erstellt und anschließend gespeichert, freigegeben und präsentiert werden<sup>6</sup>. Damit ermöglicht diese Funktion eine durchgängige Berichterstellung im Q-Cockpit ohne auf PowerPoint zugreifen zu müssen.

Die *Personalisierung* erlaubt die anwenderbasierte Speicherung der Konfigurationen von Kernfunktionen. So können beispielsweise eine personalisierte Start-Ampelansicht festgelegt oder Alerts mit selbst definierten Onlineberichten dargestellt werden.

*Kenngrößen-Cluster* kapseln die Komplexität der Vielzahl an Kenngrößen in einer zielgruppenspezifischen Struktur. So adressiert z. B. der Balanced-Score-Card-Clus-

<sup>6</sup>Dabei wird nur die Abfolge von den URLs der Darstellungen gespeichert und wiedergegeben.

ter die Unternehmensführung und ordnet die Kenngrößen nach Finanzen, Kunden und internen Prozessen.

Die *Schnellnavigation*, ähnlich einer Kommandozeile, erlaubt es, Kenngrößen, Dimensionen, Ausprägungen und Filter als Texteingabe zu formulieren, um eine Expertenansicht zu erzeugen. Dadurch erübrigt sich eine aufwendige Konfiguration einer Ansicht über die graphische Bedienoberfläche.

### 3.3.4 Evolution des Geschäftsprozesses

Die Einführung der LS3 hat die Verwendung der Kernfunktionen beschleunigt und vereinfacht. Dadurch wurde der neue Q-Cockpit-basierte Qualitätsmanagementprozess weiter optimiert und die Nutzung des gesamten Potentials der Applikation erleichtert. In der Folge stieg die Nutzung der Applikation auf mehr als das 1.5-fache im Vergleich zur LS2 während der Trend der Ressourcenoptimierung im Qualitätsmanagement weiter anhielt.

Die LS3 optimierte und komplettierte die bestehenden Kernfunktionen und verankerte damit die Nutzung des Potentials der Applikation im Unternehmen. Zusätzlich konnte durch die optimierte Ressourcenverteilung die kontinuierliche Weiterentwicklung bis zur nächsten Serie an Q-Cockpit Entwicklungsprojekten sichergestellt werden.

## 4 Strategischer Applikations-Lebenszyklus

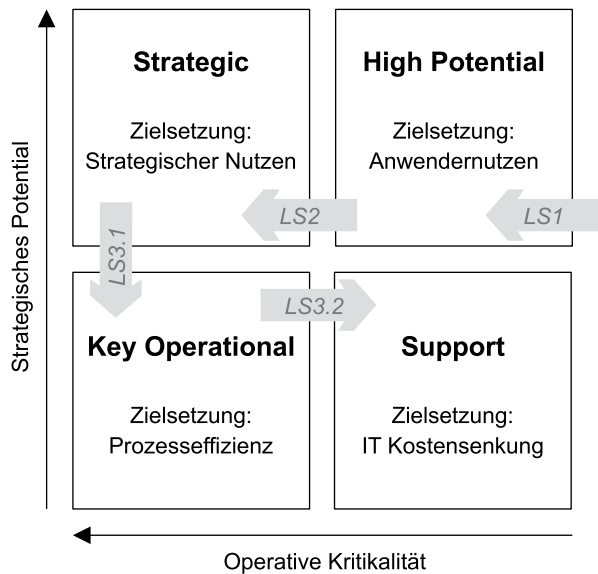
Die langfristige Beobachtung des Q-Cockpits ermöglicht die Untersuchung der Entwicklung der operativen und strategischen Bedeutung der Applikation im Unternehmen. Dabei wird deutlich, dass pro LS jeweils Ziele aus einer Kategorie (siehe Kapitel 1) dominierten. Ein von (Ward und Peppard 2002) ursprünglich zur Klassifikation eines Applikationsportfolios entwickeltes Modell eignet sich, um den Wandel der verfolgten Zielsetzung und die damit einhergehende Veränderung der Bedeutung einer Applikation in ihrem Lebenszyklus darzustellen. Das Modell spannt anhand der zwei Achsen „Strategisches Potential“ und „Operative Kritikalität“ vier Klassen auf, welche die in Kapitel 1 vorgestellten dominanten Ziel-Kategorien in der Software-Entwicklung widerspiegeln. In Abb. 2 ist der Evolutionspfad des Q-Cockpits anhand der drei Leistungsstufen visualisiert. Der Übergang nur zur jeweils benachbarten Applikationsklasse bedeutet, dass die Anwender pro LS nur mit einer Form der Veränderung konfrontiert waren.

So unterstützte die intuitive Aufteilung des Q-Cockpit-Projekts in applikationsklassenspezifische Teilprojekte sowohl die Staffelung multipler Zielsetzungen als auch eine Geschäftsprozessoptimierung im Rahmen der Veränderungsfähigkeit.

Um diese Vorteile geplant in Softwareprojekten reproduzieren zu können, wurde aus den LS der generische „Strategische Applikation Lebenszyklus“ (SALZ) abgeleitet (siehe Tab. 5).

Der SALZ bietet ein übergeordnetes Rahmenwerk für die strategische Ausrichtung der Schritte im *operativen* Applikationslebenszyklus nach ITIL (OGC 2002): Anforderungserhebung, Design, Entwicklung, Einführung, Betrieb, Optimierung.

**Abb. 2** In Anlehnung an das Applikations-klassenmodell nach (Ward und Peppard 2002)



Das Modell unterstützt eine methodenunabhängige (sowohl sequentielle als auch agile Methoden) und projektübergreifende Zieldefinition und Anforderungspriorisierung für Softwareprojekte.

Bei einem evolutionären Einsatz (stufenweise Entwicklung und Einführung) werden Teilprojekte gebildet, in denen jeweils eine SALZ-Phase mit dem operativen Applikationslebenszyklus abgebildet wird. Jede SALZ-Phase kann nur dann begonnen werden, wenn ihre Anforderungen vollständig erfüllt sind (siehe Tab. 5). Entspricht ein Teilprojekt den Erfolgskriterien einer Evolutionsphase, kann die folgende Phase angestoßen werden. Ist das Teilprojekt erfolglos, kann es mit neuen Prämissen wiederholt oder als Gesamtprojekt neu evaluiert werden.

Denkbar ist auch der revolutionäre Einsatz des SALZ (Big Bang). Dabei kann das Modell zur Vollständigkeitsprüfung und zur Schulungsplanung herangezogen werden, während mehrere SALZ-Phasen in einem Projekt umgesetzt werden und dabei bewusst die Veränderungsfähigkeit im Unternehmen herausgefordert wird.

## 5 Kritische Erfolgsfaktoren für den SALZ

Soll eine Applikation zukünftig eine strategische Rolle einnehmen, erfordert das bei konsequenter Umsetzung den Ausbau des unterstützten Geschäftsprozesses zu einem Alleinstellungsmerkmal für das Unternehmen. Dieser Prozess muss unter Einsatz der Applikation den industrieweiten Grad an Effizienz und Effektivität übertreffen. Bei der BMW Group wurde mit dem Q-Cockpit der Qualitätsmanagementprozess nachhaltig verbessert. Dabei waren für den Erfolg der evolutionären Entwicklung und Einführung der Applikation folgende Faktoren ausschlaggebend:

**Tab. 5** Strategisches Lebenszyklusmodell einer Applikation

Evolutionsphase	Zielsetzung	Voraussetzung	Angestrebter Applikationsumfang	Angestrebtes Anwenderverhalten	Angestrebte Prozessveränderung	Erfolgskriterium
Anwenderphase	Applikation im Geschäftsprozess etablieren	Geschäftsprozessverständnis	„Quick-Win“ Anwenderfunktionen	Eigeninitiative Nutzung aus direktem Mehrwert	Nutzen für bestehende Arbeitsschritte stiften	Etablierte Applikation
Strategiephase	Strategische Verbesserung des Geschäftsprozesses	Etablierte Applikation	Strategische Anwenderfunktionen	Nutzung neuer Potentiale, um Effektivität zu steigern	Evolution des Geschäftsprozesses durch Anwender	Effektiver Geschäftsprozess
Effizienzphase	Geschäftsprozess beschleunigen	Etablierte Applikation + Definierter Geschäftsprozess	Spezifische Geschäftsprozessunterstützung	Einheitliche Nutzung eines Geschäftsprozesses	Erhöhter Reifegrad	Effizienter Geschäftsprozess
Resourcenphase	Applikationsressourcen optimieren	Keine Geschäftsprozessänderungen	Wartungsarme Architektur/ Abschaltung <sup>a</sup>	Kostenbewusste Nutzung/ Migration <sup>a</sup>	Prozessveränderungen vermeiden <sup>b</sup>	Kostengünstige Applikation

<sup>a</sup>Wenn eine vergleichbare Applikation in einer anderen Phase verfügbar ist

<sup>b</sup>Applikationsbezogen; Veränderung durch andere Applikation möglich

### 5.1 Mit der Unterstützung vorhandener Prozesse beginnen

In der Anwender-Phase des SALZ geht es darum, eine Applikation unter den Mitarbeitern zu etablieren. Zielsetzung ist es dabei, ihre bestehenden Geschäftsprozesse besser als bisherige Software zu unterstützen und sie dadurch zu einem Wechsel zu bewegen. Eine gleichzeitige Softwareeinführung und Veränderung der Prozessabläufe (z. B. neue integrierte Workflows) erschwert die Wiedererkennung bestehender Prozessschritte und erschwert die Akzeptanz. Neue Prozessabläufe sollten deshalb erst mit einer etablierten Applikation in der Strategie-Phase umgesetzt werden (siehe Funktionsbeschreibung der LS1).

### 5.2 „Quick Wins“ priorisieren

Wird eine Geschäftsprozessunterstützung in mehreren Teilschritten umgesetzt, sollten zuerst die „Quick-Wins“ realisiert werden. Dies sind die Inhalte und Funktionen, die das beste Verhältnis zwischen Entwicklungsaufwand (möglichst gering) und Anwendernutzen (möglichst hoch) darstellen. Diese Priorisierung fördert die Akzeptanz des Systems und reduziert gleichzeitig das Risiko hoher Entwicklungskosten, falls das Veränderungspotential sich nicht bestätigt und das Projekt beendet wird (siehe Funktionsumfang der LS1 in Tab. 3).

### 5.3 Erfolgskriterium freiwillige Nutzung

Der beste Indikator, ob eine Applikation oder eine Funktion den Anwendern einen Mehrwert bietet, ist der Umfang, zu dem sie freiwillig eingesetzt wird. Die unverbindliche Nutzung dient in der Anwender-Phase als Erfolgskriterium, während sie in der Strategie-Phase die Innovation des Geschäftsprozesses erlaubt. Erst wenn der Mehrwert bestätigt und die gewünschte Prozessveränderung vollzogen sind, kann in der Effizienz-Phase in Erwägung gezogen werden, die Nutzung der Applikation für ausgewählte Geschäftsprozesse verpflichtend einzuführen. Verfrühte Applikations- und Prozessstandards lähmen die anwendergetriebene Prozessinnovation, machen den Applikationsnutzen intransparent und fördern die parallele Entwicklung von Schatten-IT-Lösungen<sup>7</sup> (siehe freiwillige Nutzung des Q-Cockpits in Kap. 3).

### 5.4 Mitarbeiter integrieren, aber nicht überfordern

Essentiell für einen hohen Nutzungsgrad ist die aktive Einbindung der Anwender in die Entwicklung. Um den zusätzlichen Aufwand für die Anwender einzugrenzen, müssen dabei eine intuitive Bedienung, ein konkreter Anwendernutzen und eine angemessenen Frequenz der Releases sichergestellt werden. Gerade in den ersten beiden SALZ-Phasen gilt es, die Applikation auf Anwender wirken zu lassen und sie nicht zusätzlich mit häufigen Änderungen zu überfordern (siehe auch Anzahl Releases der LS in Tab. 1).

### 5.5 Refactoring der Architektur

Die Applikationsarchitektur sollte entsprechend neuer Zielsetzung und Anforderungen, z. B. bei einem Phasenwechsel im SALZ, flexibel angepasst werden (Refactoring), um zu vermeiden, dass sie zum Flaschenhals für eine Evolutionsphase wird. Dies ist ein notwendiger Prozess der evolutionären Entwicklung und kein Zeichen unzureichender Entwicklungsarbeit früherer Phasen (siehe auch Technologieänderungen des Q-Cockpits in Tab. 1).

## 6 Übertragbarkeit des SALZ und Ausblick

Mit diesem Beitrag wurde der SALZ vorgestellt, der auf den Erkenntnissen aus dem BMW Q-Cockpit-Projekt beruht. Mit dem Modell werden die grundlegenden Zielsetzungen einer strategischen Applikation in klar abgegrenzte Teilprojekte gegliedert, die jeweils einen unabhängigen Mehrwert für das Unternehmen liefern und eine Geschäftsprozessevolution erlauben. Indem sie die Veränderungsfähigkeit des Unternehmens berücksichtigt und ein BITA auf Applikationsebene sichergestellt, macht sich diese Vorgehensweise die Vorteile des organischen Wachstums auf kontrollierte Art und Weise zu nutze.

<sup>7</sup>Ursachen und Folgen von „Schatten-IT“ werden in folgenden weiterführenden Artikel erörtert (Zimmermann und Rentrop 2012; Györy et al. 2012).

Der Erfolg des Q-Cockpits ist ein Beispiel wie mit einer evolutionären Einführung eines neuen Informationssystems entlang des SALZ Geschäftsprozesse anwendergetrieben und nachhaltig verbessert werden können.

Künftig gilt es, die Nützlichkeit des SALZ in weiteren evolutionären Entwicklungsszenarien, auch außerhalb der Business Intelligence Domäne, zu verifizieren. So könnten durch weitere Forschung z. B. ERP Systeme, mobile Kundenapplikationen oder auch Legacy-Systeme abhängig von ihrer Kritikalität und strategischen Bedeutung in dem Modell verortet werden, um deren nächste Evolutionschritte zu bestimmen. Zudem besteht die Möglichkeit, die Phasen des Modells auch im revolutionären Einsatz zu erforschen, bei dem gleichzeitig mehrere Veränderungen für Anwender, Geschäftsziele, Geschäftsprozesse und IT-Systeme berücksichtigt werden müssen (z. B. bei der Einführung neuer Systeme um Prozesseffizienz zu steigern).

## Literatur

- Bayer M. (2010) BMW gewinnt Business-Intelligence-Auszeichnung, <http://www.computerwoche.de/a/bmw-gewinnt-business-intelligence-auszeichnung,1931207>. Zugegriffen: 20. Feb 2014
- Cleven A (2011) Exploring patterns of business-it alignment for the purpose of process performance measurement. Proceedings of the European Conference on Information Systems (ECIS), Helsinki
- Ewusi-Mensah K (2003) Software development failures: anatomy of abandoned projects. MIT Press, Cambridge
- Györy A, Cleven A, Uebernickel F, Brenner W (2012) Exploring the shadows: IT governance approaches to user-driven innovation. Proceedings of the European Conference on Information Systems (ECIS), Barcelona
- Hallerstede SH, Danzinger F, Bullinger AC, Möslin KM (2011) Akzeptanzorientiertes application life-cycle management. HMD Prax Wirtschaftsinform 278:30–40
- Office of Government Commerce (2002) Best practice for application management. ITIL The key to managing IT services. TSO, London
- Österle H., Blessing D (2005) Ansätze des Business Engineering. HMD Prax Wirtschaftsinform 241:7–17
- Rupp C, Simon M, Hocker F (2009) Requirements engineering und management. HMD Prax Wirtschaftsinform 267:94–103
- Scheeg J, Pilgram U (2003) Integrierte Kostenbetrachtung für IT-Produkte. HMD Prax Wirtschaftsinform 232:89–97
- Ward J, Peppard J (2002) Strategic planning for information systems. 3. Aufl. John Wiley & Sons, Chichester
- Zimmermann S, Rentrop C (2012) Schatten-IT. HMD Prax Wirtschaftsinform 288:60–68