



Kernkompetenzzentrum
Finanz- & Informationsmanagement



Projektgruppe
Wirtschaftsinformatik

Diskussionspapier

"Zählen, wiegen, messen" - IT-Transformationen erfolgreich steuern

von

Regina Pflieger

in: HMD - Praxis der Wirtschaftsinformatik, 51, 2, 2014, S.164-174

Die finale Publikation ist verfügbar unter:
<http://dx.doi.org/10.1365/s40702-014-0016-1>

WI-450

Universität Augsburg, D-86135 Augsburg
Besucher: Universitätsstr. 12, 86159 Augsburg
Telefon: +49 821 598-4801 (Fax: -4899)

Universität Bayreuth, D-95440 Bayreuth
Besucher: F.-v.-Schiller-Str. 2a, 95444 Bayreuth
Telefon: +49 921 55-4710 (Fax: -844710)



Springer: DOI: 10.1365/s40702-014-0016-1

Titel:

„Zählen, wiegen, messen“ – IT-Transformationen erfolgreich steuern

Kurztitel:

IT-Transformationen erfolgreich steuern

Zusammenfassung:

IT-Transformationsprojekte sind die Mammuts unter den IT-Vorhaben. Anhand der etablierten Steuerungsmechanismen eines erfolgreichen IT-Transformationsprojekts im Finanzdienstleistungssektor zeigt dieser Erfahrungsbericht, welche Herausforderungen an eine zentrale Projektsteuerung bestehen und warum diese gegenüber föderalen Ansätzen vorteilhaft ist. Konkrete Praxisbeispiele verdeutlichen dabei, warum Kriterien wie intersubjektive Vergleichbarkeit oder Aggregations- bzw. Disaggregationsfähigkeit elementar wichtig sind, welche Dimensionen und Indikatoren sich zur Steuerung bewährt haben und wie eine Operationalisierung auch ohne teure Spezialsoftware gelingt.

Gliederung:

1. Eigenschaften und Herausforderungen von IT-Transformationen
2. Die typische Struktur einer IT-Transformation
3. Schlüsselanforderungen an eine erfolgreiche Steuerung
4. Zählen, wiegen, messen: Kernindikatoren der erfolgreichen Steuerung
5. Steuern mit Bordmitteln: Die richtige Tool-Unterstützung
6. Literatur

Stichwörter:

Eigenschaften einer IT-Transformation, Struktur einer IT-Transformation, zentrale Projektsteuerung, Bewertungsdimensionen, Steuerungsparameter, intersubjektive Vergleichbarkeit, Aggregationsfähigkeit

1. Eigenschaften und Herausforderungen von IT-Transformationen

Ausgangspunkt und Basis dieses Erfahrungsberichts ist die IT-Transformation eines Finanzdienstleisters, dessen historisch gewachsene und durch Unternehmenszukäufe stark heteroge-

ne Anwendungslandschaft durch zahlreiche Schnittstellen zwischen teilweise veralteten Systemen gekennzeichnet war. Ziel der Transformation war die Konsolidierung und Homogenisierung dieser siloartigen Anwendungslandschaft durch konsequente Ausrichtung an serviceorientierten Architekturen (SOA) über die Einführung einer eigenentwickelten Middleware. Zudem erfolgte die Konsolidierung sämtlicher Kernbanksysteme, die vorher teils auf unterschiedlichen Plattformen liefen, auf SAP Standardanwendungen. Die IT-Transformation startete im Jahr 2008 und konnte zum September 2013 erfolgreich abgeschlossen werden.

Im Allgemeinen stellen IT-Transformationen wie die eben beschriebene eine große Herausforderung für Unternehmen dar. Aspekte, die als Ursache für das Scheitern von (IT-) Projekten aufgeführt werden, gewinnen vor dem Hintergrund dieser Mammut-Projekte weiter an Bedeutung. Zu nennen sind Themen wie die Komplexität von Projekthinhalten, unklare Rollenverteilungen, Anforderungen und Ziele, mangelhafte Kommunikation oder fehlendes Projektmanagementwissen und -methodik auf Führungsebene [Wieczorrek und Mertens 2008]. Bedingt durch diverse Eigenschaften von IT-Transformationsprojekten, werden einige der genannten Gefahren in diesem Projektumfeld verstärkt. IT-Transformationen sind Projekte, die die Definition dieses Begriffs im Sinne eines einmaligen Vorhabens mit Zielen, beschränkter Laufzeit, beschränktem Budget und Ressourcen weit ausdehnen (vgl. [Schulte-Zurhausen 2010], [Wieczorrek und Mertens 2008]). Projektlaufzeiten erstrecken sich oftmals über mehrere Jahre und das finanzielle Projektvolumen liegt meist im zwei bis dreistelligen Millionenbereich. Eine hohe Anzahl betroffener Geschäftsanwendungen innerhalb der Anwendungslandschaft wie auch eine große Zahl beteiligter interner wie externer Mitarbeiter sowie Dienstleister und Lieferanten prägen das Projektbild und tragen zum hohen Risiko dieser Vorhaben bei.

Um den Gefahren effektiv entgegen zu treten, kann eine zentrale Projektsteuerung den Schlüssel zum Erfolg darstellen. Begründet liegt dies in mehreren Faktoren: Ein wesentliches Argument sind die starken Abhängigkeiten zwischen den zahlreichen Einzelprojekten einer IT-Transformation. Einzelne Vorhaben und Teilschritte müssen für den Erfolg der Gesamttransformation jederzeit aufeinander abgestimmt sein, um einerseits vorhandenes Synergiepotenzial zu erkennen und zu nutzen und andererseits Dominoeffekte, die beispielsweise durch eine zeitliche Verzögerung von Teilschritten entstehen können, frühzeitig zu erkennen und zu vermeiden (vgl. kritischer Pfad [Wieczorrek und Mertens 2008]). Ein weiterer kritischer Faktor sind Budgetmehrbedarfe. In Folge des hohen Projektvolumens führen bereits kleine Zielabweichungen in Einzelprojekten zu hohen nominalen Mehrbedarfen. Daher ist, vor allem hinsichtlich der aggregierten Sicht aller Abweichungen über alle Teilprojekte, eine

rigide Budgetsteuerung erforderlich. Nicht zuletzt genießen Projekte dieser Größenordnung und Relevanz für das originäre Geschäft von Unternehmen (vgl. operationelle Risiken von Systemausfällen) eine hohe Sichtbarkeit im Unternehmensvorstand, Aufsichtsrat, aber auch gegebenenfalls bei externen Organen, wie beispielsweise der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) oder der Bundesbank (BuBa) im Falle von IT-Transformationen im Bankensektor. Die daraus resultierende Aufmerksamkeit hinsichtlich des Projektstatus und -erfolgs erfordert ein konsistentes und anschlussfähiges Reporting, das jederzeit und unter Umständen auch sehr kurzfristig und ohne hohe Abstimmungsbedarfe aussagefähig sein muss. Föderale Strukturen reichen aufgrund der genannten Eigenschaften von IT-Transformationen für diesen geforderten Gesamtüberblick nicht aus. Eine Gesamtbudgetsicht, die Organisation der gesamten Projektressourcen oder die Steuerung von Abhängigkeiten zwischen den einzelnen Teilprojekten kann nur durch eine zentrale Instanz ganzheitlich vorgenommen und erfasst werden. Abbildung 1 fasst die Eigenschaften von IT-Transformationen nochmals zusammen und zeigt die damit verbundenen Herausforderungen an eine zentrale Projektsteuerung auf.

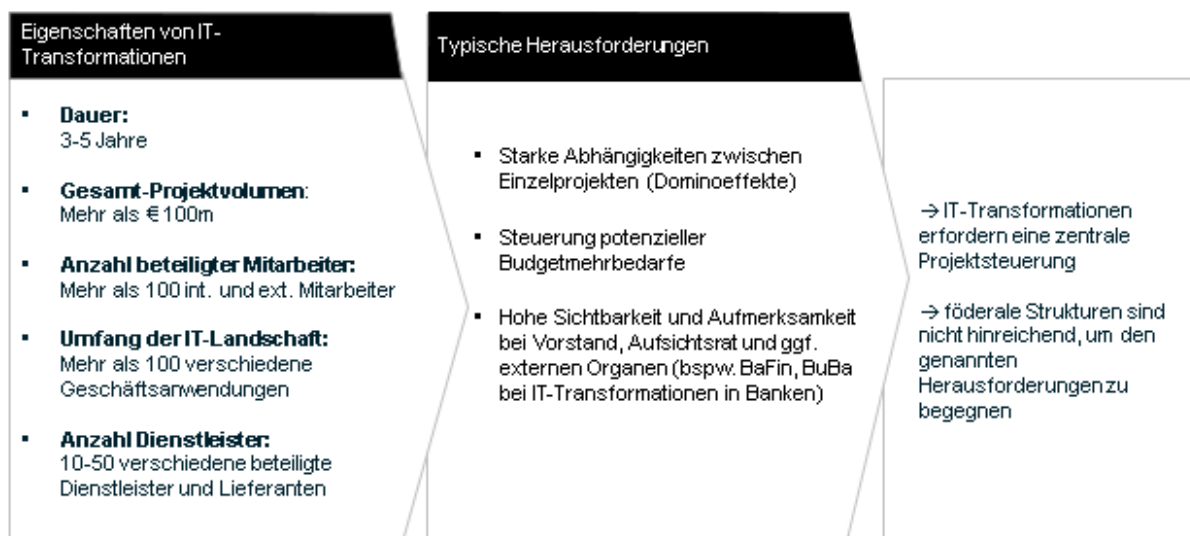


Abbildung 1: Eigenschaften von IT-Transformationen

2. Die typische Struktur einer IT-Transformation

Um im Weiteren die Anforderungen an eine erfolgreiche Projektsteuerung herauszuarbeiten, wird zunächst die typische Struktur von IT-Transformationen anhand des dem Beitrag zugrundeliegenden Beispiels vorgestellt (vgl. Abbildung 2).

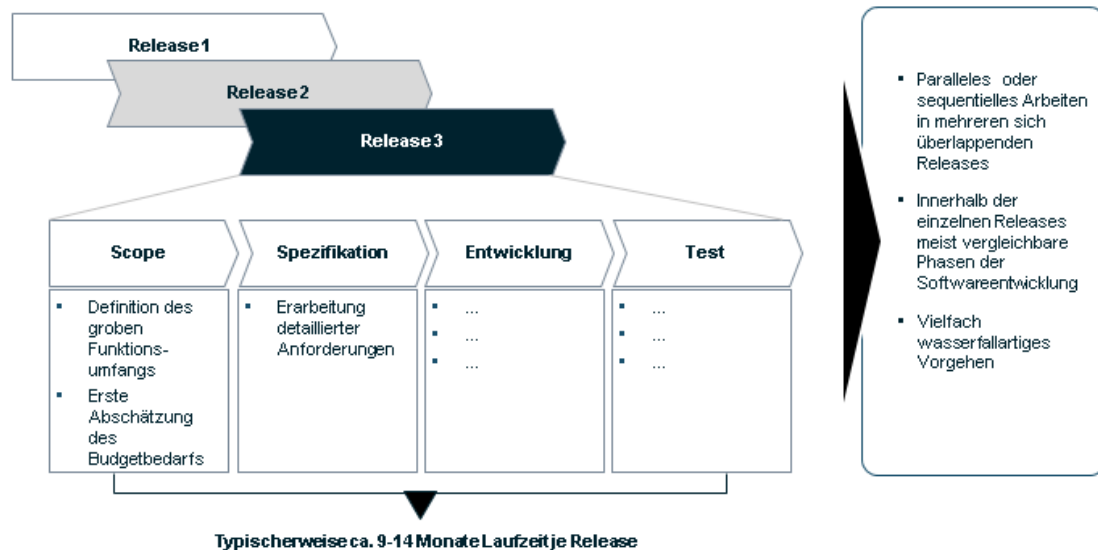


Abbildung 2: Typische Struktur und Vorgehensweise bei IT-Transformationen

Das IT-Transformationsvorhaben wurde in fünf Teilabschnitte zerlegt, um dadurch die Komplexität und damit auch das Risiko, das mit einer sehr langen Projektlaufzeit einhergeht, zu verringern. Die fünf Teilabschnitte, im Folgenden als Releases bezeichnet, wurden zeitlich sequentiell, bzw. stellenweise überlappend durchgeführt. In jedem Release waren Teilprojekte des Gesamtvorhabens gebündelt. Ein Release hatte dabei in der Regel eine Laufzeit zwischen neun und vierzehn Monaten und lieferte ein Ergebnis, wie zum Beispiel die Einführung der Middleware als Teilschritt zur Gesamttransformation. Die Gestaltung der Releases erfolgte nach den Phasen der Softwareentwicklung, meist vergleichbar für alle Releases, die sich in die Scope-Phase (Abgrenzungsphase), die Spezifikationsphase, die Phase der Entwicklung und schließlich des Tests gliederten. Dabei wurde vielfach ein an das Wasserfallmodell angelehntes Vorgehen gewählt (vgl. Vorgehensmodelle in der IT-Entwicklung, [Ruf und Fittkau 2007]).

In der Scope-Phase erfolgte die erste Schätzung des groben Funktionsumfangs eines Releases auf Basis von „Top Level Requirements“ (TLRs). Zudem wurde eine erste Abschätzung des Budgetbedarfs erhoben. In der darauffolgenden Spezifikationsphase erfolgte die weitere Detaillierung der Anforderungen resultierend in sogenannten „Mid Level Requirements“ (MLRs), wie auch die Detaillierung des Budgets. In beiden Phasen waren sowohl Verantwortliche der Fachseite wie auch der IT-Seite beteiligt, um die Projektanforderungen aus beiden Perspektiven zu erstellen und zu validieren. Diese duale Besetzung der Teams, durch Fach- und IT-Seite wurde auch auf Ebene der Leitungsfunktionen in den Teil- und Unterprojekten sowie auf oberster Ebene der Projektsteuerung vorgenommen. In der Entwicklungsphase erfolgte die Umsetzung teils in agiler Entwicklung innerhalb mehrerer IT-

Entwicklungsplattformen. Diese bündelten die Entwicklungsarbeiten an einer Applikation wie beispielsweise der Middleware oder aller Applikationen eines Funktions- oder Anwendungsbereichs, wie zum Beispiel Front-Office oder Back-Office-Systeme. Die Entwicklungsplattformen waren den jeweiligen Teilprojekten eines Releases orthogonal zugeordnet, d.h. eine Plattform lieferte gleichzeitig an mehrere Teilprojekte, wobei jede Plattform einen IT-seitigen Leiter („Platform-Head“) hatte, der die Zulieferungen an die Teilprojekte koordinierte und verantwortete. Das Ergebnis der Entwicklung in den Teilprojekten wurde im Rahmen der Testphase wiederum fach- wie auch IT-seitig hinsichtlich Funktionalität und Erfüllung der Anforderungen überprüft. Die duale Besetzung der Teams und Leitungsfunktionen durch Fach- und IT-Seite, wie auch das iterative Vorgehen in den einzelnen Releases sicherte dabei, dass die formulierten Anforderungen inhaltlich sowie qualitativ dem entsprachen, was tatsächlich von der Fachseite benötigt wurde und die gelieferten Softwareartefakte diese wiederum auch erfüllten.

3. Schlüsselanforderungen an eine erfolgreiche Steuerung

Welche Anforderungen muss eine erfolgreiche Projektsteuerung nun erfüllen, um die vorgestellte IT-Transformation über alle Releases hinweg zentral zu führen? Zwei wesentliche Elemente haben sich hierzu aus dem vorliegenden Praxisbeispiel abgeleitet. Einerseits gab es verschiedene Anspruchsgruppen auf verschiedenen Ebenen, innerhalb und außerhalb des Projekts und/oder des Unternehmens, die Informationen zum Projektstatus einforderten. Andererseits gab es verschiedene Projektbeteiligte, die wiederum diese Information zum Projektstatus abgaben. Nachfolgende Beispiele sollen die daraus folgenden wesentlichen Anforderungen an die Projektsteuerung verdeutlichen: *Der CIO muss dem Vorstand eine Einschätzung dazu geben, ob eine hinreichende Testabdeckung zum geplanten Go-Live Termin erreicht werden kann. Ein Release-Manager hingegen muss auf Wochenbasis einschätzen können, ob alle Projekte eines Releases rechtzeitig zum Integrationstest die erforderliche Software liefern können.* Status-Reports jeglicher Granularitätsebene (vgl. Teilprojekt-, Release-, oder Gesamtprojektebene) mussten folglich effizient, schnell und ohne aufwändige Abstimmungsprozesse mit vielen betroffenen Projektmitarbeitern erstellt werden können. Um das zu erfüllen mussten die Steuerungsparameter einfach zu aggregieren sein (z.B. additiv), um so jederzeit den Anspruchsgruppen Auskunft zum Projektstatus geben zu können. Wie aber sieht diese Auskunft aus, wenn unterschiedliche Projektteilnehmer diese abgeben? *Der Programm-Manager muss beispielsweise einschätzen, ob der vom Kreditrisikobereich als kritisch eingeschätzte Zustand tatsächlich besorgniserregend oder nur Ergebnis der potenziell starken Risikoaversion des*

Bereichs ist. Der Projektleiter im Bereich der Handelssysteme hingegen will verstehen, ob die optimistische Einschätzung der Entwickler über die Effizienz der Fehlerbehebung wirklich gerechtfertigt ist – oder aber vergleichsweise positiv gesehen wird, da die beteiligten Projektteilnehmer tendenziell risikofreudig sind. Die Parameter der Projektsteuerung mussten folglich so gewählt werden, dass sie eine intersubjektive Vergleichbarkeit zulassen. Das heißt anders formuliert, sie sollten eine objektive Einschätzung des Zustands ermöglichen, die nicht, beziehungsweise nur in geringem Maße von der individuellen Risikopräferenz der verantwortlichen Projektleiter bzw. -beteiligten abhängig ist. In der Praxis der Projektsteuerung findet man häufig freitextartige Statusberichte. Diese erfüllen die genannten Schlüsselanforderungen der Aggregationsfähigkeit und intersubjektiven Vergleichbarkeit nicht. Im Gegenteil: freitextartige Statusberichte können diese vielmehr erschweren. Zusammenfassungen durch dritte Personen können den Inhalt verfälschen oder wichtige Aspekte des Projektstatus werden weg gelassen. Zudem spiegeln die gewählten Formulierungen die Risikopräferenz des Erstellenden wider und lassen so keine objektive Beurteilung des Projektstatus zu. Eine Steuerung auf Basis von quantitativen Steuerungsgrößen kann diesem jedoch entgegen wirken.

4. Zählen, wiegen, messen: Kernindikatoren der erfolgreichen Steuerung

Die Erfahrungen aus der dem Beitrag zugrunde liegenden IT-Transformation bestätigen den Erfolg einer Projektsteuerung, die sich auf einige wenige, in den einzelnen Release-Phasen wiederkehrende, quantitative Steuerungsgrößen fokussiert. Drei Dimensionen haben sich dabei bewährt: Budgetverbrauch, Projektfortschritt und Projektqualität. Jede dieser Bewertungs- oder Steuerungsdimensionen wurde auf die jeweiligen Phasen innerhalb der Releases angepasst. Während der Budgetverbrauch über alle Phasen hinweg einheitlich gemessen wurde, wurden Projektfortschritt wie auch -qualität entsprechend den Ergebnissen der einzelnen Phasen angepasst. Abbildung 3 liefert eine Übersicht.

Dimensionen der Steuerung			
Release-Phasen	Budgetverbrauch	Fortschritt	Qualität
Scope	<ul style="list-style-type: none"> Laufende Zeiterfassung aller Mitarbeiter, deren geleisteter Aufwand aus dem zur Verfügung stehenden Budget zu finanzieren ist Erfassung in einer Form, die weitere Aggregation zulässt (bspw. keine physischen Stundenzettel) 	<ul style="list-style-type: none"> Messung des Fortschritts anhand des Zustands von konkreten zu liefernden Artefakten Definition von zwei bis drei Versionszuständen (bspw. Version 1.0 ist die von IT und Fachbereich abgenommene Version) hat sich bewährt 	<ul style="list-style-type: none"> Messung der Review-Monita der Lieferartefakte, deren Kritikalität und Abarbeitungszustand
Spezifikation		<ul style="list-style-type: none"> Kontinuierliche Extrapolation der „Burn-rate“ zur frühzeitigen Identifikation potenzieller Budgetüberschreitungen 	<ul style="list-style-type: none"> Frühzeitige Definition von Lieferartefakten mit sinnvoller Größe Messung der Anzahl fertiger Artefakte (statt des verbrauchten Aufwands)
Entwicklung			
Test		<ul style="list-style-type: none"> Abarbeitungszustand der Testfälle im Zeitablauf und Erfolgsmessung je Testfall 	<ul style="list-style-type: none"> Messung der Anzahl dokumentierter Fehler und Einordnung der Fehler in vordefinierte und homogene Fehlerklassen

Abbildung 3: Dimensionen der Projektsteuerung in den Release-Phasen

Der Budgetverbrauch wurde anhand der laufenden Zeiterfassung aller Mitarbeiter, deren geleisteter Aufwand aus dem zur Verfügung stehenden Budget zu finanzieren ist, gemessen. Die Erfassung erfolgte dabei in einer Form, die eine weitere Aggregation der Daten zulässt, konkret durch die Erfassung geleisteter Projektstunden, die zentral gepflegt wurden (keine physischen „Stundenzettel“). Es erfolgte zudem eine kontinuierliche Extrapolation der „Burn-Rate“ (tatsächlicher Budgetverbrauch) gemeinsam mit einem stetigen Soll-Ist-Vergleich, um potenzielle Budgetüberschreitungen frühzeitig identifizieren zu können. Der Budgetverbrauch wurde in allen Release-Phasen und über alle Releases hinweg auf gleiche Art und Weise erfasst. So war jederzeit eine Aggregation bzw. Disaggregation dieser Bewertungsdimension möglich. Die Anforderung der Aggregationsfähigkeit der Steuerungsgröße war folglich erfüllt. Durch die quantitative (vgl. Stundenanzahl) und monetäre (vgl. geschätztes Budget, tatsächlich verbrauchtes Budget) Erfassung ist zudem eine unabhängige und intersubjektiv vergleichbare Aussage möglich.

Der Projektfortschritt wurde mit Hilfe der Anzahl und des Fortschritts von zu liefernden Artefakten erfasst. In der Scope- und Spezifikationsphase erfolgte dies anhand des Zustands der zu liefernden Artefakte. Bewährt hat sich hier die Definition von zwei bis drei Versionszuständen. Die Version 1.0 ist dabei beispielsweise die vom Fachbereich und der IT abgenommene finale Version. In der Entwicklungsphase erfolgte die Messung über die Anzahl der fertiggestellten Artefakte (statt des verbrauchten Aufwands). Diese mussten dazu bereits frühzeitig und in jeweils sinnvollem Größenumfang definiert werden. In der Testphase wurde der Projektfortschritt anhand des Abarbeitungszustands der Testfälle im Zeitablauf und durch die

Erfolgsmessung je Testfall gemessen. Wiederum erfüllen die gewählten Indikatoren die Anforderungen der Aggregationsfähigkeit und intersubjektiven Vergleichbarkeit an die Projektsteuerung durch die Wahl eines quantitativen Steuerungsparameters. So gibt die Anzahl aller fertiggestellten Softwareartefakte eine von der Risikoeinstellung des Berichterstattenden nahezu unabhängige Einschätzung des Projektfortschritts, die von jedem Informationsempfänger gleich und auch unabhängig von dessen Risikopräferenz eindeutig aufgefasst werden kann (die Konsequenz aus der Information und damit Einschätzung der Situation ist dann freilich die subjektive Wahrnehmung des Entscheiders). Zudem ist, sofern sinnvoll und benötigt, eine beliebige Zusammenfassung der Daten, wie beispielsweise Anzahl der Testfälle eines Teilprojekts versus Anzahl der Testfälle aller Projekte innerhalb eines Releases, je nach Interesse und Berichtsebene des Informationsempfängers möglich.

Die Projektqualität wurde für die Phasen Scope und Spezifikation anhand der Messung der „Review-Monita“ (Beanstandungen, die im Zuge der Überprüfung gefunden wurden) der jeweiligen Lieferartefakte, deren Kritikalität und Abarbeitungszustand gemessen. Alle gefundenen Monita wurden dabei zentral erfasst, Verantwortlichkeiten festgelegt und auch der zugehörige Status dokumentiert. In der Entwicklung erfolgte die Erfassung der Qualität nur innerhalb der jeweiligen Entwicklungsplattformen über sogenannte „Bug-Tracker“. Dies diente der internen Steuerung je Plattform und wurde nicht zentral erfasst. In der Testphase erfolgte die Erhebung der Projektqualität wiederum zentral, über die Messung der Anzahl der dokumentierten Fehler und deren Einordnung in Fehlerklassen. Vorausgesetzt einer eindeutigen Definition dieser Kritikalitätsklassen, lässt auch hier die größtenteils quantitative Bewertung eine Aggregation der Ergebnisse zu und erlaubt eine intersubjektive Vergleichbarkeit der Daten.

5. Steuern mit Bordmitteln: Die richtige Tool-Unterstützung

Die Umsetzung der Projektsteuerung erfordert ein Mindestmaß an IT-Unterstützung. Dabei kann beispielsweise auf etablierte Projektmanagement-Tools am Markt zurückgegriffen werden. Alternativ sind selbsterstellte Lösungen denkbar. Erfahrungsgemäß trifft man in der Praxis jedoch oftmals auf eine Kombination aus Standardlösungen und Eigenentwicklungen, was mehrere Ursachen haben kann. Der Markt bietet eine breite Angebotspalette an Standardapplikationen mit unterschiedlichem Funktionsumfang, von spezifischen Anwendungen zur Steuerung von Teilprozessen im Projekt bis hin zu allumfassenden Tools. Da IT-Transformationen jedoch nicht „auf der grünen Wiese“ entstehen, sondern Unternehmen in der Regel Applikationen zur IT-Steuerung einsetzen (bspw. SAP im Bereich der Fakturierung und Buchhaltung oder Tools zur Anforderungsanalyse oder zum Fehlermanagement) entsteht

die Herausforderung, Standardlösungen möglichst nahtlos in die Applikationslandschaft einzufügen. Häufig ist jedoch mit „IT-Lücken“ in der Prozessunterstützung oder Funktionsüberlappungen zu rechnen, die bewertet werden müssen. Zusätzlich entstehen neben Anschaffungs- und Lizenzgebühren meist hohe Aufwände für das Customizing der Software und die Anbindung an bestehende Applikationen, sofern die erforderliche IT-Kompetenz nicht intern verfügbar ist. Nichtsdestotrotz bieten Standardlösungen durch die vorgegebenen Messgrößen und Strukturen vor allem in Unternehmen mit wenig eigenen Erfahrungswerten hinsichtlich der Steuerung von Großprojekten eine gute Anleitung und Orientierung. Eigenentwicklungen bieten hingegen vor allem den Vorteil, dass sie passgenau liefern können, was benötigt ist, und dabei meist flexibler und günstiger in bereits vorhandene Strukturen integriert werden können. Die Folge ist, dass in den meisten Unternehmen eine Mischlösung existiert, die einen unternehmens-individuellen Trade-off darstellt, um die jeweiligen Vorteile von Standardlösungen und Eigenentwicklungen zu nutzen und gleichzeitig die zugehörigen Nachteile zu reduzieren.

Im vorliegenden Beispiel fiel zu Projektbeginn die Entscheidung für eine umfassende Standardsoftware (Microsoft Office Project Server (MOPS)) zur zentralen Projektsteuerung und dem Projektportfoliomanagement der Transformation. Die Investition in eine Standardlösung war dabei rechtfertigbar durch den langen zeitlichen Horizont der Transformation und der damit verbundenen hohen Anzahl an parallel laufenden Projekten und zu steuernden Parametern, was zu der Erwartung führte, dass sich die Einführung der Standardsoftware amortisieren wird. Bereiche wie das Testmanagement (RQM, [IBM 2014]) oder Einkaufsprozesse, z.B. von externen Dienstleistern (Ariba, [ARIBA 2014]) wurden durch weitere vorhandene Standardlösungen abgedeckt. Daneben fanden sich auch eigenentwickelte Tools wie bspw. Microsoft Office (MS) Excel-Listen zum Tracking von Change Requests (formaler Beschluss, um die Rahmenbedingungen eines Projekts nachträglich an geänderte Anforderungen anzupassen, z.B. durch späte Änderungen in zu erfüllenden gesetzlichen Anforderungen) im Project Management Office. Im letzten Teil des Projekts erfolgte mit Blick auf das nahende Projektende und dem damit einhergehenden „Eindampfen“ der temporären großdimensionalen Projektstrukturen während des Transformationsprojektes, die Ablösung von MOPS durch eine eigenentwickelte Alternative. Die Umsetzung in MS-Excel wurde mit rund 100 Personentagen Aufwand von Mitarbeitern des Project Management Office realisiert. Die Wahl von MS-Excel ist dabei durch mehrere Faktoren begründet: es ist im Unternehmen vorhanden, ausreichend lizenziert, und bietet einen hinreichenden Funktionsumfang, um die Parameter zu steuern. Zudem kann es von vielen Mitarbeitern ohne zusätzliche Anwenderschulungen bedient

werden, was auch auf die Entwicklung zutrifft, für die keine weiteren (externen) Experten hinzugezogen werden mussten.

Die Projektsteuerung auf Basis des eigenentwickelten „Excel-Toolkits“ enthält drei wesentliche Elemente: eine projektspezifische Excel-Datei je Teilprojekt innerhalb der Releases, eine zentrale Excel-Datei über alle Teilprojekte eines Releases sowie eine zentrale Excel-Datei zur Budgetverwaltung eines Releases (vgl. Abbildung 4).



Abbildung 4: Übersicht „Excel-Toolkit“

Die projektspezifische Excel-Datei liefert eine detaillierte Planung und Steuerung auf Einzelprojektebene für alle Teilprojekte eines Releases. Stammdaten wie zum Beispiel die Projektklassifikationsnummer oder die Zuordnung zu einem Release wie auch die Projektklassifikation erfolgen zentral über das Projekt Management Office. Auch die beispielsweise durch die Zuordnung zu einem Release vorgegebenen Meilensteine werden durch das Project Management Office eingespeist. Die Projektleiter ergänzen die Projektplanung und aktualisieren die Daten fortlaufend. So erfolgt auch die Zeiterfassung der Mitarbeiter eines Projekts über die projektspezifische Excel-Datei. Durch Import der jeweiligen Stundenerfassung der Mitarbeiter (Projektressourcen) in die Excel-Datei wird dort der geleistete Aufwand zentral erfasst. Projektressourcen werden vorab vom Projektleiter geplant.

Die projektspezifische Excel-Datei bietet die Möglichkeit, Berichte zum Verfolgen des Budgetverbrauchs und des Fortschritts auf Projektebene zu generieren. Dazu gehören beispielsweise die Möglichkeit den Plan- mit dem Ist-Budgetverbrauch pro Projektressource bis auf Monatsebene darzustellen, oder auch den Projektfortschritt anhand der Meilensteine sowie der nötigen Lieferartefakte, wie z.B. Spezifikationen oder auch sogenannter Software Deliverable Objects (SDO), zu bewerten (vgl. Abbildung 5).

Project Status Report							
Basic Data							
Release:	2014 - Release 1	Business Area:	Finance	Project End:	31.03.2014	Go-Live:	16.02.2014
Project:	1 14 007 - New GL						
Project Lead Business:	Name Business PL			Project Lead IT:	Name IT PL		
Short Scope Description:	Merger of two general ledgers to a single ledger, thus ensuring streamlined and faster accounting and closing processes. Moreover, the advantages of New General Ledger are realized, e.g. providing data with different accounting rules in a single ledger using a single integrated Chart of Accounts.						
Overall Status Summary							
Time Status:	Green	Scope Status:	Green	Budget Status:	Green	Report Date:	20.01.2014
Report Week:	4						
Current	Last	Status Comment			Mitigation Action		
Green	Green	GoLive (system change from Classic GL to NewGL) and conversion of source systems to NewGL logic successfully implemented. Retest of open test cases triggered, bug fixing ongoing. Preparation for "Generalprobe": Weekend work approved, systems in preparation, Runbook activities finalized. NewGL trainings scheduled and documentation prepared. Detailed planning for remaining tasks 2014 including GoLive weekend finalized.			Monitor the year end process closely (currently in time).		
Detailed Statuses of involved teams (first line: Business team(s), further line(s): IT team(s))							
Business Team /	Current	Last	Status Comment			Mitigation Action	
Business Status	Green	Green	Go-Live of software successfully executed. Nevertheless, as testing was mainly focussed on Platform 2 and there are still data inconsistencies in downstream applications, it is most likely that Platform 1 will have an intensified post go-live phase!				
IT Platform 1	Green	Green	Go-Live successfully executed.				
IT Platform 2	Green	Green	GoLive of NewGL Platform 2->Platform 3 interface (incl. emergency fixes) successful; no known issues at the moment				
IT Platform 2	Green	Green	Dry Run Preparation started				
IT Platform 3	Green	Green	Comprehension questions about NewGL-4043 and defects 4030, 4032, 4034. Discussions ongoing				
IT Platform 3	Yellow	Green	Insufficient Platform Budget allocation to complete ordering systems & Clean ups. New system components needed by 15 Feb			- Clarify budget with project lead - discuss system requirements roadmap.	
IT Platform 4	Green	Green	no known issues				
Budget Overview							
EUR (gross)				PD			
Milestones							
Milestone	Baseline Date	Current Date	CE	Open Tasks	Deliverables		
Milestone - Scope Agreement	01.03.2013	28.02.2013	139	0	Total	Open	
Milestone - End of Spec (v1.0)	12.07.2013	12.07.2013	208	0	8	8	
Milestone - End of Realisation	31.10.2013	31.10.2013	1.172	3	3	2	
Milestone - End of Testing	16.02.2014	31.03.2014	1.008	423	2	2	
Milestone - Go Live	17.02.2014	16.02.2014	0	0	5	5	
Milestone - Project End	31.03.2014	31.03.2014	74	74	0	0	

Abbildung 5: Excel-Toolkit Auszug aus dem Projektstatusreport eines Einzelprojekts

Alle projektspezifischen Excel-Dateien werden durch das Project Management Office in einer zentralen Excel-Datei konsolidiert. Sämtliche Kerninformationen zu einem Projekt, wie Stammdaten, Projektrisiken und Projektstatus (inklusive der Kernindikatoren zur Steuerung) werden erfasst. Die zentrale Excel-Datei dient vor allem der Aggregation der in den projekt-

spezifischen Excel-Dateien sehr detaillierten Informationen. Ziel ist es dabei, ad hoc sowie in wöchentlich erstellten „Cockpits“ und Berichten, einen konsolidierten Status der Projekte in verschiedensten Aggregationsstufen für das Management bereitzustellen. Die zentrale Excel-Datei ist der erste Anlaufpunkt bei sich abzeichnenden Zielabweichungen, bevor in den projektspezifischen Excel-Dateien nach Ursachen gesucht wird. Sie dient damit neben der Gesamtberichterstattung auf Management Ebene auch für das Controlling im Projekt. Die zentrale Excel-Datei wird ausschließlich von Mitarbeitern des Project Management Office bearbeitet.

Ergänzend findet die zentrale Budgetverwaltung, welche wiederum ausschließlich durch das Project Management Office vorgenommen wird, in einer weiteren Excel-Datei statt. So wird ein „Single Point of Truth“ bezüglich der Budgetzahlen sichergestellt. Hauptbestandteil ist die Budgetverwaltung und -steuerung, wie zum Beispiel die Zuweisung phasenspezifischer Budgets, das Management von Change Requests sowie die zugehörige Berichterstattung.

Die vorgestellte Umsetzung einer zentralen Projektsteuerung stellt eine pragmatische Lösung aus der Praxis dar, die auf Basis der drei Bewertungsdimensionen Budgetverbrauch, Projektfortschritt und Projektqualität die erfolgreiche Steuerung einer IT-Transformation unterstützt. Die eingesetzten Steuerungsparameter erfüllen die Anforderungen der Aggregierbarkeit und intersubjektiven Vergleichbarkeit.

6. Literatur

[ARIBA 2014] Ariba, 2014, <http://de.ariba.com/>, Abruf am 24.01.2014

[IBM 2014] Rational Quality Manager (RQM), 2014, <http://www-03.ibm.com/software/products/en/ratiqualmna>, Abruf am 25.01.2014.

[Ruf und Fittkau 2007] Ruf W., Fittkau T.: Ganzheitliches IT-Projektmanagement: Wissen, Praxis, Anwendungen. Oldenbourg Wissenschaftsverlag, München 2007.

[Schulte-Zurhausen 2010] Schulte-Zurhausen, M.: Organisation. In: Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. Vahlen, München 2010.

[Wieczorrek und Mertens 2008] Wieczorrek H. W., Mertens P.: Management von IT-Projekten. Springer-Verlag Berlin. Heidelberg 2011.