

2 Begriffliche und theoretische Grundlagen

In diesem Kapitel werden die begrifflichen und theoretischen Grundlagen der Arbeit erläutert.

Zuerst widmet sich das Kapitel dem Begriff der ERP-Systeme. Diese Informationssysteme stellen oft das Herzstück in der IT-Landschaft eines Unternehmens dar. Das Kapitel geht hier insbesondere auf die Postimplementierungsphase von ERP-Systemen ein und erarbeitet die zentralen IT-bezogenen Aufgaben sowie mögliche Erfolgsmaße in dieser Phase.

Eine Herausforderung in der ERP-Postimplementierungsphase ist die Sicherstellung einer optimalen Unterstützung der Geschäftsprozesse durch das ERP-System. Hierfür sind verschiedene Fähigkeiten erforderlich, um das ERP-System effizient zu betreiben, effektiv einzusetzen und kontinuierlich an neue Geschäftsbedingungen anzupassen. Im zweiten Abschnitt dieses Kapitels wird deshalb der Begriff der IT-bezogenen Fähigkeiten aus der Perspektive der Resource Based View erläutert.

Der letzte Abschnitt betrachtet schließlich den Begriff der Reifegradmodelle und analysiert, inwieweit Reifegradmodelle als Instrument zur Entwicklung von IT-bezogenen Fähigkeiten eingesetzt werden können.

2.1 ERP-Systeme

ERP-Systeme sind heute ein zentraler Bestandteil in der IT-Landschaft von Unternehmen. In vielen Unternehmen ist das ERP-System sogar das umfassendste Informationssystem, sowohl in Bezug auf die finanzielle Investition als auch die beteiligten Mitarbeiter (Chang et al. 2008a, 929). ERP-Systeme können definiert werden als betriebliche Informationssysteme, die auf Standardsoftware basieren und einen Großteil der betriebswirtschaftlichen Funktionsbereiche in einem Unternehmen in integrierter Form unterstützen (Klaus/Rosemann/Gable 2000, 141; Schwarzer/Krcmar 2010, 153). Ihr Umfang deckt Funktionsbereiche wie die Materialwirtschaft, die Produktion, den Vertrieb und das Finanz- und Personalwesen ab (Davenport 1998, 124; Kurbel 2011, 4). ERP-Systeme sind somit in der Lage, eine Vielfalt von Ressourcen eines Unternehmens zu verwalten, wie Material, Personal, Kapazitäten, Finanzen und Information (Gronau 2010, 4). Aus technischer Sicht werden ERP-Systeme insbesondere durch eine Verarbeitung von Information in Echtzeit (Brehm 2001, 8; Davenport 1998, 124) und die Nutzung einer gemeinsamen Datenbank für alle betriebswirtschaftlichen Funktionsbereiche (Brehm 2001, 8; Davenport 1998, 124; Kurbel 2011, 227) charakterisiert.

Zwar haben ERP-Systeme einen breiten Funktionsumfang, sie können jedoch durch weitere betriebliche Informationssysteme ergänzt werden, die Spezialfunktionen für ausgewählte Bereiche bieten. Beispiele sind Informationssysteme für das Supply Chain Management oder das Customer Relationship Management (Krcmar 2010, 29). Diese Informationssysteme können Schnittstellen zum ERP-System aufweisen und so die gesamte Wertschöpfungskette eines Unternehmens unterstützen (Gronau 2010, 11).

2.1.1 *Eigenschaften von ERP-Systemen*

ERP-Systeme kennzeichnen sich durch zwei zentrale Merkmale, die neben der erstmaligen Implementierung auch das spätere Management dieser Systeme in einem Unternehmen beeinflussen. Diese Merkmale sind zum einen eine hohe *Integration* und zum anderen der Einsatz von *Standardsoftware* als Basis für ERP-Systeme.

2.1.1.1 Merkmal Integration

ERP-Systeme kennzeichnen sich durch eine hohe Integration verschiedener betriebswirtschaftlicher Funktionsbereiche und Geschäftsprozesse, die durch das System unterstützt werden (Gronau 2010, 5; Jacob 2008, 1f.; Kurbel 2011, 227; Markus/Tanis 2000, 176). Der Begriff der Integration kann unter anderem anhand ihres Gegenstands und ihrer Richtung näher beschrieben werden (Mertens 2009, 2). Der Gegenstand der Integration kann sich auf Daten, Funktionen, Prozesse, Methoden oder Programme erstrecken, wobei eine gemeinsame Datenhaltung den Mindestumfang an Integration für ein ERP-System darstellt (Gronau 2010, 5; Mertens 2009, 1).

In Bezug auf die Richtung der Integration kann zwischen einer horizontalen sowie vertikalen Integration unterschieden werden (Mertens 2009, 5): Die horizontale Integration bezieht sich auf eine Verbindung der verschiedenen betriebswirtschaftlichen Funktionsbereiche entlang der Wertschöpfungskette eines Unternehmens und spiegelt somit den Grundgedanken eines ERP-Systems wieder. Die vertikale Integration umfasst die Versorgung von Planungs- und Kontrollsystemen, z. B. ein Data Warehouse und darauf aufbauende analytische Systeme, mit operativen Daten.

Aus einer hohen Integration folgen verschiedene Implikationen. Zum einen kann sich eine Änderung in einem integrierten Informationssystem, z. B. die Änderung eines Programms oder Parameters, auf verschiedene betriebswirtschaftliche Funktionsbereiche auswirken und somit umfassende Wechselwirkungen hervorrufen (Mertens 2009, 12). Jede Änderung erfordert somit eine große Sorgfalt und ein strukturiertes Verfahren, um eine reibungslose Abwicklung der Geschäftsprozesse sicherzustellen. Zum anderen kann aufgrund der hohen Komplexität des Informationssystems eine Änderung bestehender Funktionalität meist nur von Experten durchgeführt werden, da die Auswirkungen der Änderung schwer einschätzbar sind (Gronau 2010, 8f.). Letztendlich führt eine hohe Integration auch zu einer großen Anzahl an Systemnutzern aus verschiedenen Fachbereichen, die unterschiedliche und zum Teil konkurrierende Anforderungen an das ERP-System haben. Aus diesem Grund stellt die fachbereichsübergreifende Koordination und Kommunikation ein wesentlicher Erfolgsfaktor über den gesamten ERP-Lebenszyklus dar (Grabski/Leech/Schmidt 2011, 42; Somers/Nelson 2004, 270).

2.1.1.2 Merkmal Standardsoftware

ERP-Systeme kennzeichnen sich auch dadurch, dass sie in der Regel auf Standardsoftware basieren, die anhand verschiedener Methoden an die Anforderungen eines Unternehmens

angepasst werden kann (Kurbel 2011, 233; Markus/Tanis 2000, 176; Seddon/Shanks/Willcocks 2003, 1).

Ein zentraler Diskussionspunkt in der Literatur ist die Anpassungsmethode, mit der eine Änderung oder Erweiterung des ERP-Systems durchgeführt wird. Die einfachste Methode ist die Konfiguration des ERP-Systems anhand von Parametern, die auch als *Parametrisierung*, *Konfiguration* oder als *Customizing im engeren Sinne* bezeichnet wird (Brehm/Heinzl/Markus 2001, 1; Kurbel 2011, 327). Über eine Parametrisierung kann z. B. die Unternehmensstruktur im System hinterlegt oder angepasst werden, oder es können Funktionen des Systems aktiviert oder deaktiviert werden. Der Vorteil einer Anpassung durch Parametrisierung ist, dass die Parametereinstellungen in der Regel ohne Anpassungsaufwand in neuere Versionen des ERP-Systems übertragbar sind (Kurbel 2011, 328ff.). Somit verursacht die Anpassung des ERP-Systems durch Parametrisierung bei einer Aktualisierung der ERP-Software den geringsten Aufwand. Falls nicht alle Anforderungen über eine Parametrisierung abgedeckt werden können, so stellen ERP-Systeme noch eine Reihe weiterer Methoden zur Verfügung, um das System anzupassen. Typische Methoden sind die Nutzung sogenannter User Exits zur Erweiterung des ERP-Quellcodes, die Erstellung von Eigenentwicklungen, die Schnittstellenentwicklung und die Modifikation des bestehenden ERP-Quellcodes (Brehm/Heinzl/Markus 2001, 4; Kurbel 2011, 327ff.).² User Exits sind durch den ERP-Hersteller vordefinierte Stellen im ERP-Quellcode, an denen individueller Programmcode eingefügt und der ERP-Quellcode so um unternehmensspezifische Geschäftslogik erweitert werden kann (Brehm/Heinzl/Markus 2001, 4; Kurbel 2011, 328). Da User Exits durch den ERP-Hersteller bereitgestellt werden, sind sie meist aufwärtskompatibel zu neueren ERP-Versionen, müssen jedoch bei einer Aktualisierung der ERP-Software getestet werden (Kurbel 2011, 331). Eigenentwicklungen hingegen sind durch das Unternehmen erstellte, eigenständige Programme, die in der Regel in der Programmiersprache des ERP-Systems erstellt werden und meist direkt in das System integriert sind (Brehm/Heinzl/Markus 2001, 4). Da Eigenentwicklungen jedoch oft Schnittstellen zur ERP-Standardfunktionalität haben, sind nach einer Aktualisierung der ERP-Software ein Test und eventuell eine Überarbeitung der Eigenentwicklungen erforderlich (Kurbel 2011, 331). Am aufwendigsten im Hinblick auf eine spätere Software-Aktualisierung ist eine Modifikation des bestehenden ERP-Quellcodes. Da der geänderte Quellcode durch eine Aktualisierung der ERP-Software überspielt werden kann, muss die Modifikation in der Regel manuell in die neue Version übertragen werden (Kurbel 2011, 331).

Typ und Umfang der verwendeten Anpassungsmethoden können somit den Aufwand für die Wartung und die Aktualisierung des ERP-Systems wesentlich beeinflussen (Beatty/Williams 2006, 108; Brehm/Heinzl/Markus 2001, 3f.; Kremer 2010, 159; Light 2001, 425). Aus diesem Grund kann eine umfangreiche Verwendung bestimmter Anpassungsmethoden die Aktualisierung des ERP-Systems, und somit auch die Nutzung neuer Funktionalität, verzögern. Da ERP-Systeme auch nach der Inbetriebnahme kontinuierlich erweitert werden, ist die sorgfältige Auswahl der Anpassungsmethoden über den gesamten ERP-Lebenszyklus von hoher Bedeutung (Markus/Tanis 2000, 195).

² Eine umfassende Aufstellung verschiedener Anpassungstypen liefern z. B. Brehm (2001, 95ff.); Brehm/Heinzl/Markus (2001) sowie Kurbel (2011, 327ff.).

Eine weitere Implikation von Standardsoftware ist, dass die Weiterentwicklung von ERP-Systemen nicht nur durch das Anwenderunternehmen erfolgt, sondern in erheblichem Umfang auch durch den ERP-Hersteller. Neben Fehlerkorrekturen im ERP-Quellcode hat dieser auch die Aufgabe, das ERP-System an technologische Entwicklungen anzupassen sowie das System um neue Funktionalitäten zu erweitern (Brehm 2001, 54). Der ERP-Hersteller hat somit einen signifikanten Einfluss auf die Häufigkeit und den Umfang der angebotenen Software-Aktualisierungen (Nah/Faja/Cata 2001, 401). Für das Anwenderunternehmen bedeutet dies eine erhebliche Abhängigkeit vom ERP-Hersteller im Hinblick auf die Weiterentwicklung des Systems. Aus diesen Gründen wird die Beziehung zum ERP-Hersteller sowie dessen Unterstützung als wesentlicher Erfolgsfaktor über den gesamten ERP-Lebenszyklus hinweg gesehen (Somers/Nelson 2004, 267).

2.1.2 ERP-Lebenszyklus

Der Lebenszyklus von Anwendungen wird durch Lebenszyklus-Modelle beschrieben, die verschiedene Phasen umfassen. Der Lebenszyklus einer Anwendung umfasst ganz allgemein

„[...] die zeitliche Entwicklung von der ursprünglichen *Idee* und der *Entscheidung*, eine Anwendung zu entwerfen oder zu beziehen, über die *Entwicklung* und *Einführung* des neuen Systems bis hin zur *Wartung* und einer etwaigen *Weiterentwicklung* der existierenden und genutzten Anwendung bis zur abschließenden *Abschaffung*.“ (Krcmar 2010, 157)

Die meisten Modelle zur Beschreibung des ERP-Lebenszyklus umfassen zwischen vier und sechs Phasen.³ Eines der am häufigsten referenzierten Modelle ist das Lebenszyklus-Modell von Markus/Tanis (2000, 190ff.), das vier Phasen umfasst:

- Im Rahmen der *Chartering* Phase erfolgen vorbereitende Aktivitäten, z. B. eine Nutzenbetrachtung für das ERP-System, die Auswahl des Software-Pakets, Entscheidungen zur Projektorganisation sowie eine Genehmigung des Projektbudgets.
- Die *Project* Phase umfasst alle Aktivitäten, die erforderlich sind, um das System in einen betriebsbereiten Zustand zu überführen. Dies umfasst die Konfiguration, den Test, die Datenmigration, Schulungen und die eigentliche Inbetriebnahme des Systems.
- In der *Shakedown* Phase wird das System in den regulären Betrieb überführt. In dieser Phase werden verschiedene Nacharbeiten durchgeführt, z. B. die Behebung von Fehlern, die Nachschulung der Anwender oder eine Performanceoptimierung des Systems.
- Zuletzt folgt die *Onward and Upward* Phase. Sie ist gekennzeichnet durch eine kontinuierliche Verbesserung von Geschäftsprozessen, Technologie und Qualifikation der Anwender.

Hilfreich sind Lebenszyklus-Modelle vor allem deshalb, weil jede Phase andere Eigenschaften, Aktivitäten, Erfolgsfaktoren und Ergebnisse aufweisen kann. Lebenszyklus-Modelle

³ Beispiele hierfür sind zu finden in Ahituv/Neumann/Zviran (2002, 61ff.); Bajwa/Garcia/Mooney (2004, 83ff.); Chang et al. (2008b); Markus/Tanis (2000, 190ff.); Rajagopal (2002, 92); Ross/Vitale (2000, 235ff.).

helfen somit ERP-Systeme differenzierter und in Abhängigkeit von ihrem aktuellen Entwicklungsstand zu betrachten. Auch diese Arbeit nutzt Lebenszyklus-Modelle für eine erste Einordnung des Untersuchungsgegenstandes. Der Fokus dieser Arbeit liegt auf den Phasen nach der erstmaligen Inbetriebnahme des ERP-Systems, in dem Modell von Markus/Tanis (2000, 190ff.) entspricht dies der *Shakedown* sowie der *Onward and Upward* Phase. Vereinfacht wird der gesamte Zeitraum nach der ersten Inbetriebnahme auch mit den Begriffen *ERP-Post-Go-Live* oder *ERP-Postimplementierungsphase* bezeichnet (Brehm 2001, 2; Willis/Willis-Brown 2002, 37).

2.1.3 Aufgaben in der ERP-Postimplementierungsphase

In der ERP-Postimplementierungsphase erfolgt der Betrieb sowie die kontinuierliche Anpassung und Erweiterung des ERP-Systems. Betrachtet man die Aufgaben in der ERP-Postimplementierungsphase, so sind hier seit Anfang der 2000er Jahre verschiedene Vorschläge für eine Aufgabenstrukturierung entstanden, die im Folgenden diskutiert werden.

Brehm (2001, 205ff.) untersuchte anhand von vier Fallstudien die ERP-Postimplementierungsphase und schlägt eine Kategorisierung der Aufgaben in dieser Phase anhand der Kategorien *Anwenderunterstützung*, *Fehlerbehebung*, *funktionale Änderungen und Erweiterungen*, *Releasewechsel* und *technische Betreuung des ERP-Systems* vor. Diese Aufgaben lassen sich in anwendungsseitige und technische Aufgaben kategorisieren, wobei hierfür die Kategorie *Releasewechsel* weiter untergliedert werden muss (siehe Abbildung 3).

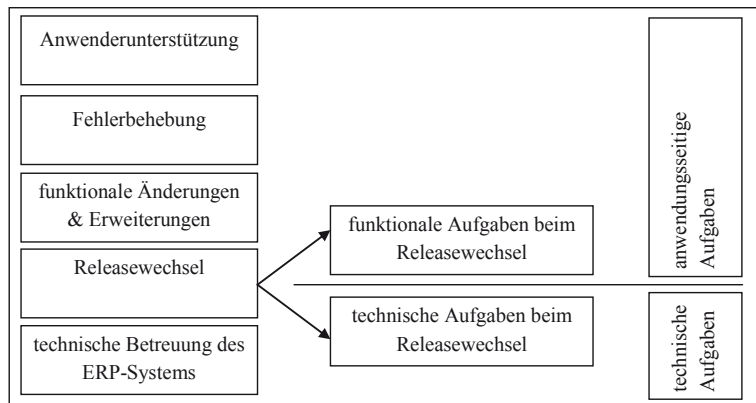


Abbildung 3: Aufgaben in der ERP-Postimplementierungsphase

(Quelle: (Brehm 2001, 208))

Eine weitere Kategorisierung liefern Nah/Faja/Cata (2001), die anhand von drei Fallstudien ebenfalls eine Reihe von Aufgaben in der ERP-Postimplementierungsphase ableiten und diese sechs Kategorien zuordnen. Die Kategorien lehnen sich an die klassische Softwarewartung an und umfassen die *korrektive*, *adaptive*, *perfektive* und *präventive Wartung* sowie die *Anwenderbetreuung*. Darüber hinaus führen die Autoren eine sechste Kategorie (externe Parteien)

ein, da der Austausch und die Zusammenarbeit mit Dritten wie dem ERP-Hersteller oder Beratungsunternehmen eine zentrale Aufgabe in der ERP-Postimplementierungsphase ist (Nah/Faja/Cata 2001, 405).

Eine dritte Kategorisierung, die ebenfalls auf Fallstudienforschung basiert, schlagen Ng/Gable/Chan (2002) vor. Diese umfasst insgesamt neun Kategorien, geht aber nicht im Detail auf die Aufgaben innerhalb einer Kategorie ein.

Für eine Zusammenführung der Aufgaben aus den Arbeiten von Brehm (2001), Nah/Faja/Cata (2001) und Ng/Gable/Chan (2002) werden im Folgenden die fünf von Brehm (2001, 205ff.) vorgeschlagenen Kategorien als Basis verwendet, da diese die heute für ERP-Systeme übliche Unterscheidung zwischen technischen Aufgaben und anwendungsseitigen Aufgaben vorsehen.⁴

Um eine eindeutige Zuordnung der Aufgaben zu ermöglichen, werden die Kategorien weiterhin wie folgend angepasst:

1. Die Kategorie *Releasewechsel* wird in *Software-Aktualisierung* umbenannt, sodass diese Kategorie auch Software-Aktualisierungen umfassen kann, die von ihrem Umfang nicht einem Releasewechsel entsprechen. Dies betrifft beispielsweise Aufgaben rund um das Einspielen von Patches des ERP-Herstellers, die Ng/Gable/Chan (2002, 105) in eine eigene Kategorie einordnen.
2. Für eine klare Trennung zwischen *technischen* und *anwendungsseitigen Aufgaben* wird die Kategorie *Software-Aktualisierung* in die zwei Kategorien *Software-Aktualisierung – technisch* und *Software-Aktualisierung – anwendungsseitig* aufgeteilt.

Unter diesen sechs Kategorien können die im Folgenden beschriebenen Aufgaben gebündelt werden.

• Anwenderunterstützung

Die Anwenderunterstützung umfasst die Unterstützung und Beratung von Anwendern bei Fragen und Problemen der Systemnutzung sowie die Schulung von Anwendern (Brehm 2001, 206; Nah/Faja/Cata 2001, 406; Ng/Gable/Chan 2002, 97ff.).

• Fehlerbehebung

Die Fehlerbehebung umfasst die Beseitigung von Fehlern in der Konfiguration des ERP-Systems, in bestehenden Anpassungen des ERP-Systems, in Systemschnittstellen sowie die Beseitigung von Anwenderfehlern, die z. B. durch eine Falscheingabe entstanden sind (Brehm 2001, 206; Nah/Faja/Cata 2001, 406; Ng/Gable/Chan 2002, 97ff.). In der Arbeit von Nah/Faja/Cata (2001, 406) ist in der Kategorie *Fehlerbehebung* auch das Einspielen von Patches bzw. Korrekturen des ERP-Herstellers vorgesehen. Da das eigentliche Einspielen von Patches jedoch eine technische Aufgabe darstellt, ist diese Aufgabe in dieser Arbeit der Kate-

⁴ So unterscheiden z.B. Ferstl/Sinz (2008, 435) den systemtechnischen und fachlichen Betrieb.

gorie *Software-Aktualisierung* – *technisch* zugeordnet. Jedoch fällt in diese Kategorie die Weiterleitung von Fehlermeldungen an den ERP-Hersteller, falls ein Fehler durch den Original-Quellcode des ERP-Systems verursacht wird (Brehm 2001, 206).

- **Funktionale Änderungen und Erweiterungen**

Diese Kategorie umfasst die Änderung und Erweiterung der Konfiguration des ERP-Systems sowie die Änderung und Erweiterung des Systems durch andere Anpassungsmethoden, wie Eigenentwicklungen oder Modifikationen (Brehm 2001, 206f.; Nah/Faja/Cata 2001, 406; Ng/Gable/Chan 2002, 97ff.). Neben der eigentlichen Konfiguration oder Entwicklungstätigkeit schließt dies auch die Durchführung von Tests mit ein (Nah/Faja/Cata 2001, 406).

- **Software-Aktualisierung – anwendungsseitig**

Diese Kategorie umfasst alle anwendungsseitigen Aufgaben, die durch eine Aktualisierung der ERP-Software anfallen. Dies sind insbesondere die Überprüfung bestehender Konfigurationen, die Überprüfung und Überarbeitung bestehender Anpassungen, die Erweiterung der Konfiguration zur Nutzung neuer Funktionalität als auch der Test der Software-Aktualisierung (Brehm 2001, 207; Nah/Faja/Cata 2001, 406; Ng/Gable/Chan 2002, 97ff.).

- **Software-Aktualisierung – technisch**

Diese Kategorie umfasst alle technischen Aufgaben im Rahmen einer Aktualisierung der ERP-Software. Dies umfasst das Einspielen von Patches oder neuen Softwareversionen sowie die Anpassungen der zugrunde liegenden Hardwarekomponenten (Brehm 2001, 207; Nah/Faja/Cata 2001, 406; Ng/Gable/Chan 2002, 97ff.).

- **Technische Betreuung des ERP-Systems**

Diese Kategorie umfasst die Betreuung der technischen Infrastruktur des ERP-Systems (z. B. Hardware, Systemsoftware, Datenbank) sowie die Aufgaben der laufenden Administration (z. B. die Überwachung der Speichernutzung, die Druckerverwaltung, das Workflow-Monitoring, die Vergabe von Berechtigungen) (Brehm 2001, 207; Nah/Faja/Cata 2001, 406). Darüber hinaus wird auch der von Nah/Faja/Cata (2001, 406) erwähnte Transport von entwickelten Objekten in das Produktivsystem der technischen Betreuung zugeordnet.

Eine Zusammenfassung der Aufgaben pro Kategorie zeigt Tabelle 1.

Bei den hier beschriebenen Aufgaben handelt es sich überwiegend um operative Aufgaben der ERP-Postimplementierungsphase, die in der wissenschaftlichen Literatur auch unter dem Begriff der *ERP-Wartung* diskutiert werden.⁵ Führungsaufgaben werden hier nur in geringem Umfang betrachtet. Nah/Faja/Cata (2001, 406) schlagen zusätzlich die Aufgabe *Koordination und Verwaltung von Dritten* vor, die als einzige Aufgabe keiner der hier verwendeten Kategorien zuordenbar ist.

⁵ Siehe hierzu Nah/Faja/Cata (2001); Ng/Gable/Chan (2002).

Kategorie	Aufgaben
Anwendungsseitige Aufgaben	
Anwenderunterstützung	- Anwenderschulung - Unterstützung und Beratung von Endanwendern bei Fragen und Problemen der Systemnutzung
Fehlerbehebung	- Behebung von Fehlern in Konfiguration, bestehenden Änderungen und Erweiterungen oder Schnittstellen des Systems - Behebung von Anwenderfehlern, z. B. durch falsche Dateneingaben - Weiterleitung von Fehlern im Original-Quellcode an den ERP-Hersteller
Funktionale Änderungen und Erweiterungen	- Änderung oder Erweiterung des Systems durch Konfiguration - Änderung oder Erweiterung des Systems durch andere Anpassungsmethoden, z. B. durch Eigenentwicklungen oder Modifikationen - Durchführung von Tests
Software-Aktualisierung – anwendungsseitig	- Überprüfung der Konfiguration im Zuge von Software-Aktualisierungen - Überarbeitung bestehender Anpassungen im Zuge von Software-Aktualisierungen, wie Eigenentwicklungen, Schnittstellen, Modifikationen - Erweiterung der Konfiguration zur Nutzung neuer Funktionalität - Durchführung von Tests
Technische Aufgaben	
Software-Aktualisierung – technisch	- Einspielen von Patches und Korrekturanleitungen des ERP-Herstellers - Einspielen von neuen Softwareversionen des ERP-Herstellers - Anpassung der Hardwarekomponenten
Technische Betreuung des ERP-Systems	- Betreuung von Hardware, Systemsoftware, Datenbank - Laufende Administration des ERP-Systems (z. B. Überwachung von Speichernutzung, Druckerverwaltung, Workflow-Monitoring, Berechtigungen) - Transport von entwickelten Objekten in das Produktivsystem

Tabelle 1: Operative Aufgaben in der ERP-Postimplementierungsphase

(Quelle: Eigene Darstellung, entwickelt in Anlehnung an Brehm (2001); Nah/Faja/Cata (2001); Ng/Gable/Chan (2002))

Eine andere Perspektive auf die Aufgaben in der ERP-Postimplementierungsphase liefern Ng/Gable/Chan (2003) und Ng/Gable (2010), die ein Prozessmodell für die Wartung von ERP-Systemen vorschlagen. Das Prozessmodell umfasst die drei Bereiche *Vorbereitung der ERP-Wartung*, *Durchführung der ERP-Wartung* und *ERP-Software-Upgrade*, die jeweils verschiedene Aktivitäten umfassen (Ng/Gable/Chan 2003, 5). Das Prozessmodell enthält auch verschiedene Führungsaufgaben, z. B. den Aufbau einer ERP-Wartungsorganisation.

Zuletzt existieren in der Literatur auch vereinzelt Arbeiten, die sich mit den Führungsaufgaben in der ERP-Postimplementierungsphase befassen. Kræmmergaard/Rose (2002) untersuchen beispielsweise die Führungsaufgaben und -kompetenzen über den ERP-Lebenszyklus hinweg und verwenden hierfür das Lebenszyklus-Modell von Markus/Tanis (2000, 189).

Kennzeichnend für die Aufgaben in der ERP-Postimplementierungsphase ist, dass diese nicht alleine der IT-Organisation zuzuordnen sind, sondern diese können neben der IT-Organisation auch auf Key User oder dritte Parteien, wie den ERP-Hersteller oder externe IT-Dienstleister, verteilt sein (Hirt/Swanson 2001, 393). So ist anzunehmen, dass die Steuerung dieser Parteien ebenfalls eine zentrale Führungsaufgabe in der ERP-Postimplementierungsphase ist.

2.1.4 *Erfolgsverständnis*

Während der Erfolg⁶ eines ERP-Einführungsprojektes anhand von Kriterien wie der fristgerechten Projektfertigstellung oder der Einhaltung des Budgets ermittelt werden kann, so gelten in der ERP-Postimplementierungsphase andere Erfolgsmaße (Markus et al. 2003, 25).

Im Folgenden werden verschiedene Erfolgsmaße in der ERP-Postimplementierungsphase vorgestellt. Die verschiedenen Vorschläge aus der Literatur werden hierbei entsprechend der gewählten Perspektive strukturiert: So kann Erfolg zum einen aus der Sichtweise des ERP-Anwenders bzw. der Geschäftssicht betrachtet werden, zum anderen können aber auch IT-bezogene Aspekte berücksichtigt werden.

2.1.4.1 Erfolg aus Anwender- und Geschäftssicht

Für den Erfolg des ERP-Systems aus Anwender- und Geschäftssicht sind in der Literatur verschiedene Erfolgsmaße zu finden.

Ein einfaches Erfolgsmaß ist die Anwenderzufriedenheit. So definiert Brehm (2001, 224) beispielsweise Erfolg in der ERP-Postimplementierungsphase als „[...] das Ausmaß der empfundenen Zufriedenheit hinsichtlich dem ERP System“, wobei er sich hier auf die Sichtweise der ERP-Anwender und des Managements fokussiert.

Andere Ansätze nutzen ein breiteres Spektrum an Erfolgsmaßen. So setzen Häkkinen/Hilmola (2008) beispielsweise im Rahmen einer Fallstudie verschiedene Dimensionen des IS Success Modells von Delone/McLean (2003) als Erfolgsmaße ein. Yu (2005) nutzt für die Messung der Effektivität von ERP-Systemen ebenfalls verschiedene Variablen, nämlich Systemstabilität, Anwenderfreundlichkeit, Datenqualität, Verbesserung der Wettbewerbsfähigkeit und das Ausmaß, in dem die Erwartungen der Anwender durch das ERP-System erfüllt wurden.

Eine dritte Gruppe von Arbeiten (z. B. Nicolaou/Bhattacharya 2008) fokussiert sich auf finanzielle Kennzahlen als Erfolgsmaß.

2.1.4.2 Erfolg aus Anwender-, Geschäfts- und IT-Sicht

Einige Ansätze zur Erfolgsmessung betrachten neben der Anwender- und Geschäftssicht auch IT-bezogene Aspekte. Ein Beispiel für dieses Erfolgsverständnis liefert Jacob (2008, 2f.):

⁶ In der Literatur werden neben dem hier verwendeten Begriff des Erfolgs auch die Begriffe Wert (Jacob 2008; Jain 2010), Performance (Nicolaou/Bhattacharya 2008) oder Effektivität (Yu 2005) verwendet.

Erfolg wird hier zum einen durch einen effizienten Betrieb des Systems bestimmt, zum anderen durch eine bestmögliche Geschäftsunterstützung. Eine weitere umfassende Aufstellung von Geschäfts- als auch IT-bezogenen Erfolgsmaßen bieten Markus et al. (2003, 30f.). Erfolg in der ERP-Postimplementierungsphase wird hier bestimmt durch:

1. das Erreichen der durch das ERP-Einführungsprojekt erwarteten Ergebnisse (z. B. eine Reduzierung der IT-Kosten oder Lagerhaltungskosten),
2. die kontinuierliche Verbesserung dieser Ergebnisse,
3. die Nutzung der durch das ERP-System bereitgestellten Daten und Funktionen zur Entscheidungsunterstützung,
4. eine einfache Einführung neuer Innovationen (z. B. ein neues ERP-Release),
5. eine Sinnhaftigkeit der ursprünglichen ERP-Implementierungsentscheidung aus Sicht zukünftiger Entwicklungen und
6. eine kürzere Implementierungszeit für zukünftige ERP-Einführungen (Markus et al. 2003, 30f.).

Ein weiterer Ansatz für die Erfolgsmessung ist das von Shang/Seddon (2002) vorgeschlagene *Enterprise System Benefit Framework*, das neben anderen von Zhu et al. (2010) für die Erfolgsmessung in der ERP-Postimplementierungsphase verwendet wird. Das Framework umfasst insgesamt fünf Nutzenkategorien sowie 25 Unterkategorien (Shang/Seddon 2002, 296 ff.). Auch hier werden neben dem Geschäftsnutzen IT-bezogene Aspekte berücksichtigt.

Neben der Auswahl eines Erfolgsmaßes stellt sich letztendlich auch die Frage, wer den Erfolg bewertet. In der Literatur ist bislang strittig, inwiefern der Erfolg durch verschiedene Akteure unterschiedlich bewertet wird. So untersuchen Sedera/Gable/Chan (2004) das Erfolgsverständnis verschiedener Mitarbeitergruppen (Management, IT-Anwender, IT-Mitarbeiter) und kommen zu dem Ergebnis, dass diese Gruppen unterschiedliche Sichtweisen auf den Erfolg haben. Eine nachfolgende Untersuchung von Ifinedo (2007, 278) kommt hingegen zu dem Ergebnis, dass es keine wesentlichen Unterschiede in der Erfolgsbewertung durch Mitarbeiter der IT-Organisation und der betriebswirtschaftlichen Funktionen gibt.

2.1.5 Zusammenfassung

Ziel des Kapitels war, einen grundlegenden Überblick über ERP-Systeme und die ERP-Postimplementierungsphase zu schaffen. Im Vergleich zu anderen Informationssystemen charakterisieren sich ERP-Systeme vor allem durch das Merkmal *Standardsoftware* sowie durch ihre *hohe Integration*. Diese Merkmale haben einen wesentlichen Einfluss auf die ERP-Postimplementierungsphase, die den Zeitraum nach der erstmaligen Inbetriebnahme des ERP-Systems umfasst.

Darüber hinaus wurden im Rahmen dieses Kapitels die IT-bezogenen Aufgaben in der ERP-Postimplementierungsphase diskutiert und zusammengefasst. Zum einen fallen in dieser Phase anwendungsseitige und technische Aufgaben auf operativer Ebene an, zum anderen auch verschiedene Führungsaufgaben, z. B. die Koordination der beteiligten Parteien.

Ein Reifegradmodell für die Bewertung und
Verbesserung von Fähigkeiten im
ERP-Anwendungsmanagement

Hecht, S.

2014, XXIII, 296 S. 48 Abb., Softcover

ISBN: 978-3-658-05522-6