

ERP-Systeme erfolgreich implementieren



Das Praxis-Handbuch für
agiles Prozessmanagement



Inhaltsverzeichnis

	Vorab	9
	An wen richtet sich das Buch?	9
	Wozu das Buch?	9
	Was ist im Buch enthalten?	9
	Worauf basiert das Buch?	10
	Was ist nicht im Buch enthalten?	10
	Wie sollte man das Buch lesen?	10
Teil I	Methodisches Vorgehen	11
1	Einleitung	13
1.1	Ein neues System	14
1.1.1	Unternehmerische Notwendigkeit: Auf zu neuen Ufern ...	14
1.1.2	Release-»Upgrades«	14
1.1.3	Drop-Ins	15
1.2	Business-Software einführen, was ist zu tun?	15
1.2.1	Änderung an der Software	16
1.2.2	Änderungen im Unternehmen	16
1.3	Das methodische Vorgehen	17
1.3.1	Warum kein Wasserfall?	17
2	Die Implementierungsmethode im Überblick	23
2.1	Phasen, Zyklen und Meilensteine	23
2.2	Arbeitspakete	25
2.2.1	Strategien	26
2.2.2	Ergebnistypen der Arbeitspakete	27
2.3	Wichtige Elemente der Methode	29
2.3.1	Geschäftsprozesse	29
2.3.2	Validierungsworkshops	30
2.3.3	Validierungsskripts	33
2.3.4	Die Gap-Liste	34
2.4	Agile Organisation und agile Arbeitstechniken	35
2.5	Das Projektteam und die Stakeholder	37
2.5.1	Stakeholder	38
2.5.2	Das Projektteam	38

3	Das Projekt	43
3.1	Die Evaluierungsphase	43
3.1.1	Projekt-Initialisierung/Grobplanung	43
3.1.2	Das Suchen	45
3.1.3	Der Kauf	52
3.1.4	Die Beratung	53
3.1.5	Projekt-Setup	55
3.2	Die Implementierungsphase	58
3.2.1	Erkundungszyklus	58
3.2.2	Ausarbeitungszyklus	60
3.2.3	Erstellungszyklus	63
3.3	Die Übergangsphase	65
3.4	Die Nutzungsphase	65
 Teil 2 Die Arbeitspakete		67
<hr/>		
4	Geschäftsprozess-Abbildung	71
4.1	Vorgehensweise	71
4.1.1	Finanzwesen	72
4.1.2	Auftragswesen	73
4.1.3	Einkauf	73
4.2	Einzelarbeitspakete	74
5	Systemdesign, Erstellung und Betreuung	81
5.1	Vorgehensweise	81
5.2	Einzelarbeitspakete	81
6	Realisieren notwendiger Anpassungen	89
6.1	Vorgehensweise	89
6.1.1	Identifikation der funktionalen Lücken (Gaps)	89
6.1.2	Bewertung der Optionen	90
6.2	Einzelarbeitspakete	91
7	Datenübernahme	97
7.1	Vorgehensweise	97
7.1.1	Einfaches Szenario	97
7.1.2	Komplexes Szenario	98
7.1.3	Konsolidierungsdatenbank	98
7.2	Einzelarbeitspakete	99

8	Validieren des Systems	103
8.1	Vorgehensweise	103
8.2	Einzelarbeitspakete	103
9	Change-Management	109
9.1	Vorgehensweise	109
9.2	Einzelarbeitspakete	110
10	Schulung	115
10.1	Vorgehensweise	115
10.2	Einzelarbeitspakete	116
11	Der Systemübergang	119
11.1	Vorgehensweise	119
11.2	Einzelarbeitspakete	119
12	Stilllegung der Altsysteme	125
12.1	Vorgehensweise	125
12.2	Einzelarbeitspakete	125
13	Projektmanagement	129
13.1	Vorgehensweise	129
13.2	Einzelarbeitspakete	130
	Stichwortverzeichnis	137

Vorab

An wen richtet sich das Buch?

Das Buch richtet sich primär an Mitarbeiter von Unternehmen, die bei der Implementierung einer neuen Software Verantwortung tragen, aber auch an Systemberater, die umfangreiche Kenntnisse in den jeweiligen Softwaremodulen besitzen, aber keine Erfahrung haben, wie man eine Systemimplementierung in der Gesamtheit erfolgreich durchführt.

Wozu das Buch?

Die Implementierung von Business-Software ist kein Geheimnis und auch kein Hexenwerk. Seit mehr als 40 Jahren gehören Softwaresysteme in der Buchhaltung und in der Steuerung von Unternehmen zum Standard. Und genauso lange werden hierfür Standard-IT-Systeme implementiert und genutzt. Dabei stellt sich jedoch die Frage, warum diese Implementierungen immer noch so kostspielig und so sehr risikobehaftet sind und wie man diese Risiken reduzieren kann. Das vorliegende Buch soll hierzu Antworten geben.

Was ist im Buch enthalten?

Betrachtet man den ganzen Lebenszyklus eines Systems, so ist die **Implementierungsphase** eingereiht in die **Evaluierungs-**, die **Nutzungs-/Optimierungs-** und die **Deaktivierungsphase**.

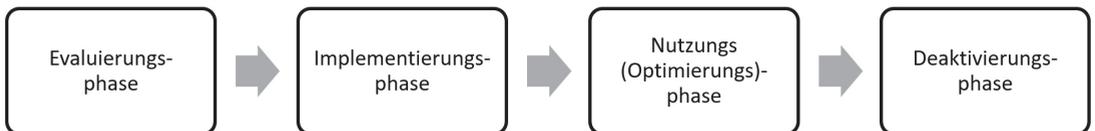


Abb. 1: Lebenszyklus eines Softwaresystems

Der Schwerpunkt des Buches sind die ersten beiden Phasen, also die Evaluierung und die Implementierung. Die Nutzungs- und Deaktivierungsphase werden nur so weit behandelt, als dass sie einen Bezug zu den anderen zwei Phasen haben.

Worauf basiert das Buch?

Die hier dargestellte methodische Vorgehensweise basiert auf den 25-jährigen Erfahrungen des Autors in der Implementierung von ERP-Software. Eingeflossen sind die Methoden *Triton Target* der ehemaligen Fa. Baan, *AIM for Business Flows* der Fa. Oracle und das von Barry W. Boehm entwickelte *Spiralmodell*.

Was ist nicht im Buch enthalten?

- Eine wissenschaftliche Abhandlung über Vorgehensmodelle und Methoden
- Eine detaillierte Beschreibung: »Was ist ERP?«
- Eine vollständige Projektmanagement-Methode
- Funktionale Konzepte
- Gewerkprojekte, also Projekte, in denen primär Software erstellt wird

Wie sollte man das Buch lesen?

Dem Autor ist bewusst, dass Sach- und Fachbücher selten von Anfang bis zum Ende gelesen werden, deshalb hier ein kurzer Hinweis:

Will man sich mit Implementierungsmethoden nur theoretisch beschäftigen, so seien die Kapitel 1, »Einleitung«, und Kapitel 2, »Die Implementierungsmethode im Überblick«, zu empfehlen.

Geht es darüber hinaus, ist man also an einer Implementierung verantwortlich beteiligt, so kommt Kapitel 3, »Das Projekt«, hinzu.

Teil 2, »Die Arbeitspakete« sollte die Projektleitung lesen, darüber hinaus dient es in erster Linie als Nachschlagewerk.

Teil I

Methodisches Vorgehen

In diesem Teil:

- **Kapitel 1**
Einleitung 13
- **Kapitel 2**
Die Implementierungsmethode im Überblick 23
- **Kapitel 3**
Das Projekt 43

Einleitung

Ein neues **Enterprise-Resource-Planning-System** (ERP-System) soll in Ihrer Firma eingeführt werden – und Sie sind dabei! Bei diesem System handelt es sich nicht um irgendeines, sondern es ist »DAS« IT-System.

Diese sind in der Regel:

- Das Finanz System mit
 - dem Hauptbuch,
 - den Nebenbüchern Debitoren, Kreditoren- und Anlagebuchhaltung
- das Materialwirtschaftssystem
- das Kunden-Management-System/CRM
- das Beschaffungssystem
- das Auftragsverwaltungssystem
- das Produktionsplanungs- und Steuerungssystem
- Lohn- und Gehalt-Software
- Personal-Management-Systeme

Nicht alle diese Systeme, wie Lohn und Gehalt, sind klassisch im Begriff ERP zusammengefasst, doch ist die hier vorgestellte Methode auch für die Implementierung solcher Systeme geeignet.

Wir sprechen hier von dem System, das die betriebswirtschaftlichen Geschäftsprozesse und Wertschöpfungen Ihres Unternehmens

- abbildet,
- dokumentiert und
- unterstützt.

Abgebildet und dokumentiert

Die Abbildung und die Dokumentation der betriebswirtschaftlichen und wertschöpfenden Prozesse ist eine Kernverpflichtung eines Unternehmens gegenüber den Eigentümern und der Gesellschaft (Staat, Finanzamt ...), die mit der Nutzung eines ERP-Systems erfüllt wird. Zwar erfolgt die Wertschöpfung im Unternehmen z.B. in der Produktion, indem neue Produkte hergestellt werden, doch die Manifestierung/Abbildung dieses Wertes erfolgt erst mit einer (Lager-)Buchung im

ERP-System. Diese und alle anderen Buchungen werden periodisch summiert und z.B. im Monatsabschluss dokumentiert. Dies ist der wichtigste Aspekt eines ERP-Systems.

Unterstützt

Darüber hinaus wird vom ERP-System erwartet, dass das System bei den betriebswirtschaftlichen und wertschöpfenden Prozessen unterstützt, sei es durch Planungs- oder Kalkulationsfunktionen oder durch die Beistellung von Informationen, um die Prozesse des Unternehmens zu steuern. Hier wären die Verkaufsstatistiken oder die Materialbedarfsplanung als Beispiel zu nennen.

1.1 Ein neues System

Beginnen wir mit dem Warum.

Warum soll ein neues System eingeführt werden? Die Antworten sind hier so vielfältig wie das Geschäftsleben selbst und dennoch lassen sich die Beweggründe gruppieren.

1.1.1 Unternehmerische Notwendigkeit: Auf zu neuen Ufern

Junge und alte Organisationen haben häufig das gleiche Problem:

Das bisher genutzte System entspricht nicht mehr den unternehmerischen Anforderungen. Dies bei jungen Unternehmen aufgrund schnellen Wachstums und veränderten Prozessen bzw. bei älteren, reiferen Organisationen, weil die Anwendung zu unflexibel ist oder durch alte Hardware und bzw. oder durch umfangreiche Anpassungen zu hohe Betriebskosten hat.

Herausforderung hier: Eine neue Software ist notwendig, und die soll natürlich alles besser können!

1.1.2 Release-»Upgrades«

Es beginnt mit der scheinbar harmlosen Information Ihres Systemlieferanten: »Ihre Version ist aus dem Support, Sie müssen auf eine aktuellere Version upgraden.«

Nun gibt es häufig das Argument: »Die Software läuft seit Jahren problemlos und den Support haben wir sowieso nur selten gebraucht, wir können damit noch lange arbeiten.« Dies ist eine Fehleinschätzung!

Betrachtet man die starke Verquickung von Hardware, Betriebssystemen und Anwendungssoftware, so ist dieses Aussitzen und Abwarten nicht ratsam! Ein banaler Ausfall und Austausch einer einfachen Netzwerkkomponente kann zu einem

notwendigen Betriebssystem-Upgrade führen, die dann nicht mehr mit der Anwendungssoftware kompatibel ist, was wiederum zum Systemstillstand führt.

Dieses Risiko sollte man bei einem unternehmenskritischen System nicht eingehen!

Mit der Einsicht in die Notwendigkeiten des Upgrades ist leider auch häufig die Erkenntnis verbunden, dass die im Betrieb eingesetzte Software-Version so stark angepasst ist, dass ein automatisches Upgrade mit den Tools des Systemlieferanten nur sehr selten, eingeschränkt, bis gar nicht möglich ist. Es ist also kein »Upgrade«, sondern eine Neuinstallation notwendig. Dies besitzt den Vorteil, dass man hier auch die Chance hat, alte Zöpfe (Funktionen, die man eigentlich nicht mehr braucht, aber das EDV-System verlangt) abzuschneiden und zusätzliche, in der neuen Version vorhandene Funktionen wie zum Beispiel das Kundenmanagementsystem zu nutzen.

Herausforderung hier: Die neue Version soll natürlich besser sein, aber die gewohnten Abläufe müssen wie bisher funktionieren.

1.1.3 Drop-Ins

In der globalisierten Welt werden Unternehmen permanent gekauft, konsolidiert und eingegliedert – oder weiterverkauft. Das Eingliedern führt bei betroffenen Unternehmenseinheiten häufig dazu, dass auf Druck der neuen »Muttergesellschaft« Prozesse vereinheitlicht und verschlankt werden müssen. Dabei wird auch häufig vorgegeben, mit welchem Business-System zukünftig gearbeitet wird, und dieses ist dann nicht immer das vorhandene.

Herausforderung hier: Das neue System wird in die Unternehmenseinheit hineingedrückt. Eine optimale Unterstützung für die Unternehmenseinheit ist hinter der Vereinheitlichung der Prozesse nachrangig.

1.2 Business-Software einführen, was ist zu tun?

Die Tätigkeiten, die im Rahmen einer ERP-Implementierung zu erledigen sind, lassen sich in zwei Bereiche einordnen.

1. Änderung an und in der Software
2. Änderungen im Unternehmen

Beide Tätigkeiten erfordern Ressourcen und haben somit Auswirkung auf die Kosten der Implementierung und auf die sich ergebenden Betriebskosten.

Dabei ist es wichtig zu verstehen, dass das Implementierungsprojekt immer beide Bereiche umfasst und ein erfolgreiches Projekt sich dadurch auszeichnet, hier die richtige Balance gefunden zu haben.

1.2.1 Änderung an der Software

Beginnen wir mit dem weniger Komplexem, der Software. Die Aktivitäten an der Software lassen sich in zwei Gruppen grob zusammenfassen,

1. der Konfiguration, also dem Einstellen der vorhandenen Funktionalität und
2. den Anpassungen (Berichte, Belege, Schnittstellen, Funktionen).

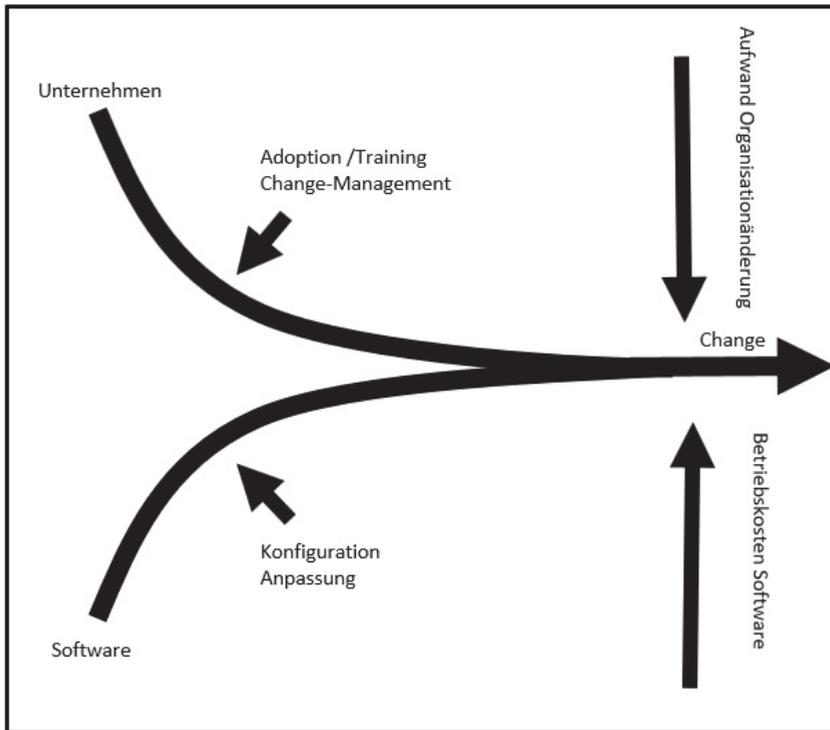


Abb. 1.1: Änderungsprozess

Diese Aktivitäten sind bei korrekter Aufnahme der Anforderungen durch erfahrende Berater in der Regel risikoarm durchzuführen. Falls es hier zu einer Kostenexplosion kommt oder Termine nicht gehalten werden können, ist dies in der Regel ein Zeichen dafür, dass nicht die richtige Balance zwischen den Änderungen an der Software und am Unternehmen gefunden wurde. Frühzeitiger Indikator ist die Gap-Liste (siehe Abschnitt 2.3.4 »Die Gap-Liste«).

1.2.2 Änderungen im Unternehmen

Die Aktivitäten im und am Unternehmen sind wesentlich komplexer. Eine neue Software verändert mit ihren neuen, anderen und ungewohnten Funktionen und

Prozessen die Arbeitsabläufe von fast allen Mitarbeitern. Das heißt, alle Mitarbeiter im Betrieb müssen sich umstellen. Sie werden teilweise ihre gewohnten Abläufe und somit Know-how verlieren und sie müssen die neuen Abläufe lernen. Dies führt insbesondere bei den erfahrenen Mitarbeitern mit viel Know-how nicht immer zu Begeisterungstürmen, denn wer fängt schon gern wieder bei null an und sei es nur in der Nutzung des IT-Systems. Hier können auch leicht Ängste oder eine Verweigerungshaltung entstehen.

Dies zu erkennen ist eine wichtige Voraussetzung, um die Implementierung erfolgreich durchzuführen, denn eine neue Software lässt sich nicht gegen die Mitarbeiter einführen.

Im Einzelnen lassen sich die Änderungsaktivitäten in

- Verstehen und Annehmen der neuen Geschäftsabläufe
- Training der Projektkernteams
- Training der End-User

gliedern.

Bei großen Veränderungen z.B. dem Wechseln von einer dezentralen zu einer zentralen Anwendung ist es empfehlenswert, diesen Prozess mit erfahrenen Beratern aus dem Themengebiet Change-Management begleiten zu lassen.

1.3 Das methodische Vorgehen

Obwohl in der Einleitung steht, dass dieses Buch keine wissenschaftliche Abhandlung über Vorgehensmodell und Methoden darstellt, so ist es doch sinnvoll, die hier beschriebene Vorgehensweise methodisch abzugrenzen. Mit methodischer Vorgehensweise beschreibe ich die Art und Weise (Methode), wie man zu einem angestrebten Ziel gelangen kann (Vorgehen).

Wie im Titel des Buches angekündigt, möchte ich hier eine Methode für eine »agile« Vorgehensweise vorstellen und diese von dem üblichen »Wasserfall-Modell« abgrenzen.

1.3.1 Warum kein Wasserfall?

In der Implementierungsphase wurde in der Vergangenheit und wird auch jetzt noch von vielen Systemhäusern und von vielen Kunden bei Eigenimplementierung eine klassische Wasserfallmethode mit den Einzelschritten

- Projektdefinition und Anforderungsanalyse,
- Entwurf mit dem Systemdesign,
- Implementierung mit Programmierung und Systemaufbau,

- Systemtest und
 - Betrieb und Wartung
- angewandt.

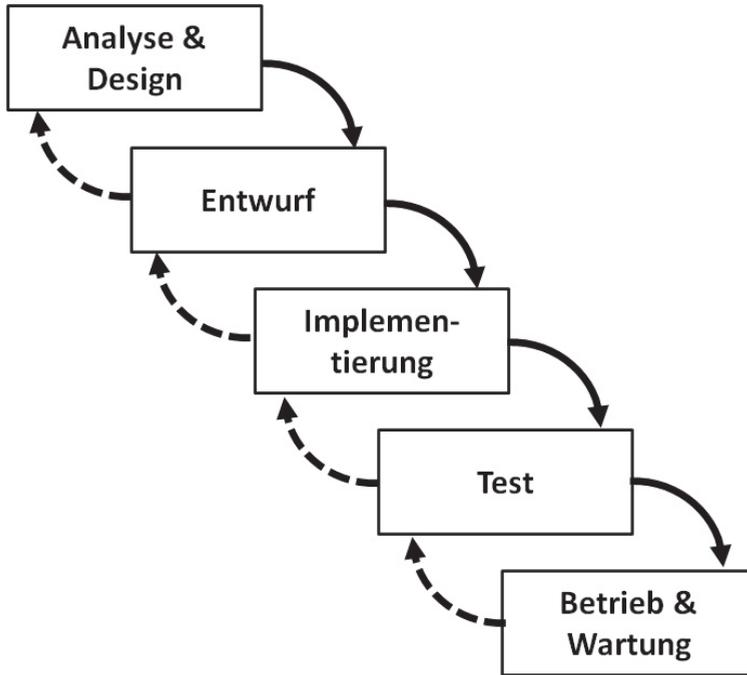


Abb. 1.2: Wasserfallmethode

In diesen Phasen wird ein System Schritt für Schritt erstellt bzw. implementiert, das den funktionalen Anforderungen, beschrieben in einem Lasten- oder Pflichtenheft, entspricht.

Diese Vorgehensweise hat allerdings einige Nachteile:

- Es gelingt selten, bereits zu Projektbeginn alle Anforderungen endgültig und theoretisch im Detail festzulegen. Daher besteht das Risiko, dass die fertiggestellte Software bzw. das implementierte System nicht den tatsächlichen realen Anforderungen entspricht. Um dem zu begegnen, wird oftmals ein unverhältnismäßig hoher Aufwand in der Analyse- und Entwurfsphase betrieben, ohne letztlich das oben genannte Risiko vollständig zu eliminieren.
- Die Wasserfall-Methode erlaubt es zwar, im laufenden Projekt Änderungen vorzunehmen (Change Requests), doch dies hat Grenzen, insbesondere dadurch, dass erst am Ende des Projekts das Ergebnis »vorzeigbar ist« und von dem Anwender beurteilt werden kann. Die fertiggestellte Software bildet folg-

lich nicht den aktuellen, sondern in der Regel den (ggf. fehlerhaften) Anforderungsstand zu Projektbeginn wieder. Da größere Softwareprojekte meist auch eine sehr lange Laufzeit haben, kann es vorkommen, dass eine neue Software bereits zum Zeitpunkt ihrer Einführung inhaltlich veraltet ist.

Diese Methode war und ist weiterhin sinnvoll, wenn ein hohes Maß an spezifischen Anforderungen bzw. Zusatzprogrammierungen zu erwarten ist, also der Gedanke der Standardsoftware in den Hintergrund tritt.

Die Alternative

Durch den erheblichen Funktionszuwachs der Standardsoftwaresysteme und den Möglichkeiten, durch Konfiguration diese Funktionen an die Anforderungen des Unternehmens anzupassen, ist es in der heutigen Zeit möglich, den Anteil der individuell erstellten Anpassungen wesentlich zu reduzieren. Dies ist deshalb wichtig, da eine Anpassung nicht nur hohe Investitionen bedeutet, sondern auch, weil sich dadurch die Wartungs- und Updatekosten des neuen Systems deutlich erhöhen.

Um zu gewährleisten bzw. zu unterstützen, dass die individuell erstellten Anpassungen so gering wie möglich gehalten werden, wird in diesem Buch eine Methode vorgestellt und beschrieben, die die funktionalen Möglichkeiten der Software in den Mittelpunkt stellt, die vom Rapid Prototype Ansatz abgeleitet ist und sich somit von der klassischen Wasserfall-Methode deutlich absetzt.

Abgeleitet vom Spiralmodell Barry W. Boehms¹ durchläuft die Methode drei Zyklen, in denen jeweils Prototypen mit unterschiedlichem Reifegrad erstellt werden.

Dabei wird der Prototyp in Test-Workshops (VWS) am Ende jedes Zyklus verifiziert. Die Workshops sind nummeriert (von 1 bis 3) und stellen den jeweiligen Reifegrad da.

Was ist daran agil?

»Agil« ist in aller Munde. Kaum eine IT-Abteilung existiert, die nicht vorgibt, »agil« zu arbeiten, so wie es sich auch kein IT-Experte leisten kann, nicht schon »agil« gearbeitet zu haben.

Hinterfragt man aber, was genau »agil« an ihrem Vorgehen ist, so gibt es häufig eine große Vielzahl von Statements.

Von »Scrum !!!«, »Pflichtenhefte brauchen wir nicht mehr« bis zu »wir brauchen keinen Plan, wir machen Sprints« reichen die Antworten. Dabei ist weder Scrum

1 Barry W. Boehm: A Spiral Model of Software Development and Enhancement. In: IEEE Computer. Vol. 21, Ausg. 5, Mai 1988, S. 61–72.

Stichwortverzeichnis

A

- Abhängigkeit 132
- Ablage
 - Projektdokumentation 57
- Altdaten-DB
 - migrieren 127
- Altsystem
 - archivieren 128
 - Stilllegung 55, 125
- AN.010 93
- AN.020 93
- AN.030 94
- AN.040 95
- AN.050 95
- AN.060 95
- AN.070 95
- AN.080 95
- AN.090 96
- AN.100 96
- AN.110 96
- Anbieter 49
 - Business-Software 45
 - Vertrag 53
- Änderungsanforderung 79
- Anpassung 89
 - Design 95
 - Kostenschätzung 94
 - Umsetzungsdesign 95
- Arbeitspaket 68, 133
 - Ergebnistyp 27
- Arbeitstechnik 35
- Archivsystem
 - bereitstellen 128
 - individuelles 128
- Ausarbeitungszyklus 24, 25, 60
 - Abschluss 32
 - Prototyp 61
 - Validierungsskript 34
- Ausschreibung 46

B

- Backlog 35
- Benutzerhandbuch 118

- Berater 40
- Beratung 53
 - Kauf 52
- Berichtswesen 134
- Betriebsführung
 - Strategie 82
- Board 36
- Budget 20, 45, 135

C

- Change-Management 97, 109
 - Tools 111
- Cloud 50
 - Infrastructure as a Service 51
 - Platform as a Service 51
 - Software as a Service 50
- CM.010 111
- CM.020 111
- CM.030 113
- CM.040 113
- CM.050 113
- CM.060 114
- Consultant 40
- Cut
 - harter 121
 - weicher 121

D

- Daily 35
- Datenabgleich 101
- Datenbank
 - Design 95
- Datenbereinigung 100
- Datenkonvertierung
 - Standards festlegen 101
 - Umgebung vorbereiten 101
- Datenmigration 85
 - Durchführung 102
- Datenmodell 96
- Datenübernahme 25, 54, 64, 97
 - Datenbereinigung 54, 97, 100
 - manuelle 101
 - Strategie 100

DAV.010 126
DAV.020 127
DAV.030 127, 128
DAV.040 128
Deltaanalyse 127
Demo 52
Demo-System 59
Deployment 96
Design
 Anpassungen 95
 Review 95
Dokumentenmanagement 27
DUE.010 100
DUE.020 101
DUE.030 101
DUE.040 101
DUE.050 101
DUE.060 101
DUE.070 101
DUE.080 102
DUE.100 102

E

Einflussmatrix 112
Einschränkung 132
End-User
 Schulungen 117
Entwicklung 86
Entwicklungsumgebung 95
Erfahrung 20, 53
Ergebnistyp
 Abnahme 29
 Dokument 27
 Softwarekonfiguration 29
 Source-Code 29
 System 28
 Workshop 29
Erkundungszyklus 24, 25, 58
 Abschluss 31
Erste Sichtung 45
Erstellungszyklus 24, 25, 63
 Abschluss 32
ETL 98
Evaluierungsphase 30, 34, 43

F

Finanzstruktur 77
First-Level-Support 123
Fragenkatalog 47
Funktionale Lücke 89

G

GA.010 76
GA.020 76
GA.030 77
GA.040 77
GA.050 77
GA.060 77
GA.070 77
GA.080 78
GA.090 78
GA.100 78
GA.120 79
GA.130 79
GA.140 79
GA.150 79
GA.160 80
GA.170 80
GA.180 80
GAP-Liste 16, 34, 60, 61, 62
 erstellen und pflegen 93
Genehmigungsverfahren 132
Gesamtsystemtest 102
Geschäftsprozess 29, 71
Geschäftsprozessablaufbeschreibung 48
Grobplanung 43, 56

H

Handlungsmandat 132
Harter Cut 121
High-Level-Lösungsdokument 78
High-Level-Solution-Dokument 80

I

Implementierung 23
Implementierungsphase 24, 58
Implementierungszeit 44
In-Class-Training 117
Infrastructure as a Service (IaaS) 51
Infrastruktur 29, 58, 134
Installationsroutine 96
Installationstest 107
Integrationstest 107
Integrationsvalidierungsskript 106

K

Kauf
 Beratung 52
 Software 49, 52
Kerngeschäftsprozessbeschreibung
 Definition 76
 Überprüfung 77

- Key-User 38
 - Kommunikation 115
 - Schulung 115, 117
- Kickoff 58
- Kommunikation 135
- Kommunikationsplan 58, 111
- Kommunikationswerkzeug 113
- Konvertierungsprogramm 101
- Kostenschätzung
 - Anpassungen 94

- L**
- Lastenheft 46
- Lebenszyklus
 - Deaktivierungsphase 24
 - Evaluiierungsphase 30, 34, 43
 - Implementierungsphase 23, 24, 58
 - Nutzungsphase 65
- Lenkungsreis 41
- Lücke
 - funktionale 89

- M**
- Mapping 101
- Meilenstein 23, 25
- Modultest 106
- Modulvalidierungsskript 106
- MoSCoW 90

- N**
- Notfallplan 123
- Nutzungsphase 65

- O**
- Organigramm 56
- Organisation 29

- P**
- Parallelbetrieb 121
- Phase 23
 - Evaluiierungsphase 43
 - Projektmanagement 129
 - Wasserfall 18
- Planung
 - Übergang 65
- Platform as a Service (PaaS) 51
- PM.000 131
- PM.010 132
- PM.020 132
- PM.030 133
- PM.040 133
- PM.050 134
- PM.060 134
- PM.070 134
- PM.080 135
- PM.090 135
- PM.100 135
- PM.110 135
- PM.120 136
- PM.130 136
- Produktionsbetriebsmanagement 87
- Produktionsstart 124
 - Abnahmekriterien definieren 122
- Produktionssupport-Konzept 122, 124
- Produktionssystem 86
 - bereitstellen 124
- Produktionssystemlandschaft 86
- Produktionssystemverfügbarkeit 124
- Produktionswartung 122
- Projekt
 - Infrastruktur 58
 - Initialisierung 43
 - Kickoff 58
 - Setup 55
- Projektabgrenzung 131
- Projektanlage 133
- Projektabschlussbericht 136
- Projektauftrag 131
- Projektauswirkung
 - Ausarbeitung 113
- Projektcontrolling 135
- Projektdokumentation
 - Ablage 57
- Projekt-Kickoff 113
- Projektkommunikation 114, 135
- Projektleitung 40, 55
- Projektlieferung 131
- Projektmanagement 129
- Projektmanagementorganisation 134
- Projektorganisation 56, 132
 - Key-User 21, 38
 - Lenkungsreis 41
 - Stakeholder 37
- Projektplan 56
- Projektsteuerung 135
- Projektsystemlandschaft 84, 86
- Projektteam 38
 - Schulung 58, 117
- Projektumfang 131
- Projektumsetzungsstrategie 111
- Projektziel 131
- Prototyp 59
 - Ausarbeitungszyklus 61
- Puffer 135

Q

Quality Gate 21, 31

R

Read-Only-Modus 127
 Referenzkarte 118
 Referenzsystem 86
 Rolle 57
 Rücktrittsrecht 53

S

Sandbox 85
 SB.010 82
 SB.020 83
 SB.030 84
 SB.040 86
 SB.050 86
 SB.060 87
 Schulung 29, 55, 86, 115
 am Arbeitsplatz 117
 durchführen 118
 End-User 117
 In-Class 117
 Key-User 117
 Projektteam 58, 117
 Zeitplan 117
 Schulungsbedarf 117
 Schulungsmaterial 118
 Schulungsstrategie 56, 116
 Schulungsumgebung 118
 Scrum-Board 135
 Second-Level-Support 123
 Sichtung
 erste 45
 Software
 Kauf 49, 52
 Softwaremodul
 Auswahl 59
 Solution-Review 78
 Stakeholder 38, 111
 Stakeholder-Analyse 111
 Stakeholder-Management 111
 Standard-Archiv-System 127
 Stilllegung 24
 Altsystem 55, 125
 Stilllegungskonzept 127
 Strategie 22, 26, 56
 Betriebsführung 82
 Datenübernahme 100
 Geschäftsabbildung 76
 Projekt 132

Projekto Umsetzung 111
 Realisierung von Anpassungen 93
 Systemübergang 120

Suchen

 System 45
 SUE.020 122
 SUE.030 122
 SUE.040 123
 SUE.050 124
 SUE.060 124
 SUE.070 124
 SUE.080 124
 SV.010 105
 SV.020 106
 SV.030 106
 SV.040 106
 SV.050 106
 SV.060 107
 SV.070 107
 SV.080 107
 SV.090 107
 SV.100 108
 SV.110 108
 SV.120 108
 SV.130 108
 System

 Validieren 103
 Systemadministration 87
 Systemarchitektur
 Übersicht 83
 Systemdesign
 Erstellung und Betreuung 81
 System-Installation 53
 Systemkonfiguration 77
 Systemlandschaft
 Analyse 79
 Systempartner 49
 Systemübergang 119
 Systemübergangsstrategie 120
 Systemvalidierungsskript 106
 Überarbeitung und Ergänzung 108
 Systemverfügbarkeit 79
 Systemwartung 124

T

Taylor'n 22
 Tenancy 51
 Test 55
 Testen
 Migration je Objekt 102
 Testpläne 33, 101
 Test-Validierung 102

- Validieren des Systems 103
- Validierungsfälle 34, 103
- Validierungsskript 32, 33, 106
- Validierungsworkshop 25, 30, 59, 64
- Testfall
 - konkreter 34
- Testmanagement
 - Standards 105
 - System 105
 - Tools 105
- Testplan 101
 - abstrakter 33, 106
- Teststrategie
 - Festlegung 105
- Third-Level-Support 123
- Timebox 91
- Tool 76
- TR.010 116
- TR.020 117
- TR.030 117
- TR.040 117
- TR.050 117
- TR.060 118
- TR.070 118
- TR.080 118
- Transaktionsmenge pro Zeiteinheit 79

U

- Übergang
 - Planung 65
- Übergangsphase 65
- Übergangsplan 123
- Übernahmestrategie 100
- Umfang 44
- Umsetzungsdesign
 - Anpassungen 95
- Umsetzungsplan 113
- Unternehmensstruktur 77

- Upgrades 14
- User-Acceptance-Test (UAT) 108

V

- Validieren
 - System 103
- Validierung 86
- Validierungsfall 34
- Validierungsskript 33
- Validierungsstrategie 56
- Validierungsworkshop 25, 30
- Validierungsworkshop 1 31, 59, 78
 - Abnahme 59
- Validierungsworkshop 2 32, 107
 - Abnahme 62
- Validierungsworkshop 3 32, 63, 108
 - Abnahme 64
- Validierungsworkshop-1-Umgebung 77
- Vertrag 52
 - Anbieter 53
- Video 118
- Vorgehensweise 44

W

- Wasserfall 17, 19
- Weicher Cut 121
- Workshop 1 45

Z

- Zielsetzung 43, 55
- Zugriffrechtsprofil 80
- Zyklus
 - Ausarbeitungszyklus 24, 32, 34, 60
 - Erkundungszyklus 24, 31, 58
 - Erstellungszyklus 24, 32, 63
 - planen 60
 - Übergangsphase 65