www.hybrid-eichhörnchen.de

CUSTOM CODE LIFECYCLE MANAGEMENT

Ganzheitliche Steuerung und Verwaltung kundeneigener Entwicklungen der SAP Business Suite. ITIL in der SAP-Anwendungsentwicklung.

14.07.2014

Peter Weigel
Hyazinthenstraβe 6
D-06122 Halle an der Saale

Telefon: +49 170 5337567
E-Mail: peter.weigel@hybrid-eichhörnchen.de
Web: www.hybrid-eichhörnchen.de

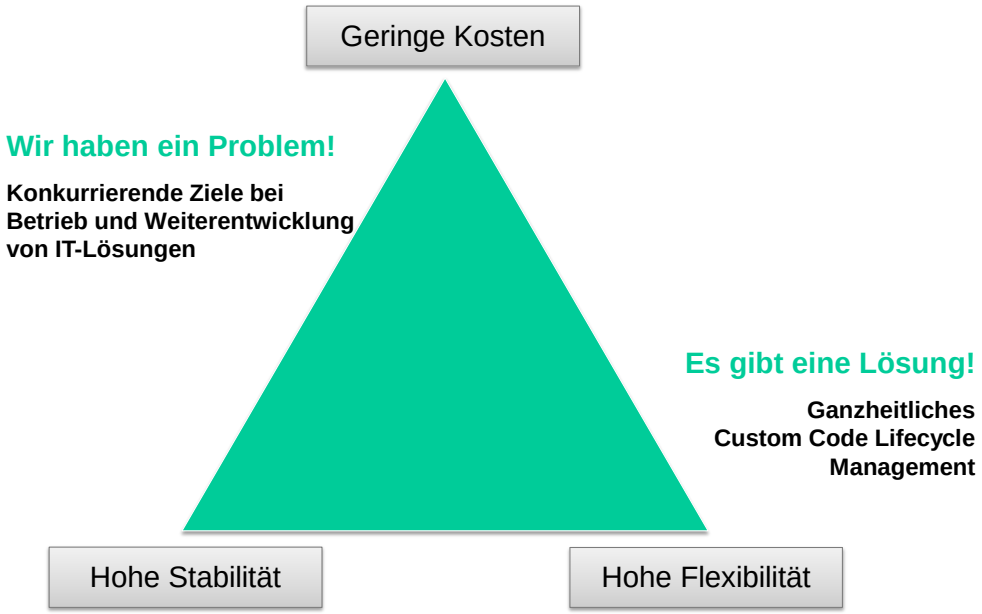
Inhalt

1	**Einleitung**	**4**
1.1	**Teil 1**	**4**
1.1.1	Motivation	4
1.1.2	Versionshistorie	6
1.1.3	Ausgangssituation	6
1.1.4	Zielsetzung	7
1.1.5	Abgrenzung	7
1.2	**Teil 2**	**8**
1.2.1	Literatur	8
1.2.2	Rechtliche Hinweise	8
1.2.3	Kontakt	8
1.2.4	Download	9
2	**Anwendungsentwicklung**	**10**
2.1	**Interview mit einem fiktiven Entwicklungsleiter**	**10**
2.2	**Theorie und Praxis eines fiktiven Entwicklers**	**13**
2.3	**Mission einer fiktiven Anwendungsentwicklungsabteilung**	**14**
3	**IT-Dienstleistungen (IT Services)**	**18**
3.1	**IT-Dienstleistung und Wertdefinition (IT Service Definition)**	**18**
3.2	**Komposition von IT-Dienstleistungen (IT Service Composition)**	**22**
3.3	**IT-Dienstleister (IT Service Provider)**	**25**
3.4	**Steuerung und Verwaltung von IT-Dienstleistungen (IT Service Management)**	**28**
4	**Bausteine von IT-Dienstleistungen (IT Service Assets)**	**31**
4.1	**Dienstleistungsbaustein "Prozesse" (IT Service Asset "Prozesses")**	**32**
4.2	**Dienstleistungsbaustein "Informationen" (IT Service Asset "Information")**	**35**
4.3	**Dienstleistungsbaustein "Wissen" (IT Service Asset "Knowhow")**	**36**
4.4	**Dienstleistungsbaustein "Anwendungen" (IT Service Asset "Applications")**	**40**
5	**Zweck von IT-Dienstleistungen (IT Service Utility)**	**49**
5.1	**Anforderung (Requirement)**	**50**
5.2	**Entwurf (Design)**	**52**
5.3	**Implementierung (Build)**	**57**
5.4	**Test (Test)**	**61**
5.5	**Auslieferung (Deployment)**	**64**
5.6	**Betrieb (Operation)**	**69**
5.7	**Optimierung (Continual/Continuous Improvement)**	**74**
5.8	**Strategie (Strategy)**	**77**
6	**Vertiefungsrichtungen und Erfolgsfaktoren**	**81**
6.1	**Custom Code Lifecycle Management**	**82**
6.2	**Werkzeuge der Anwendungsentwicklung**	**82**
6.3	**Richtlinien und Vorlagen für Konzeption, Entwicklung und Dokumentation**	**83**
6.4	**Entwicklungsberatungsleistungen**	**83**
6.5	**Produktisierung und Produktarchitektur**	**86**
6.6	**Spezialisierung, Professionalisierung und Standardisierung**	**89**
6.7	**Steuerung und Verwaltung der Anwendungsentwicklung**	**91**
6.8	**Standort- & Zielbestimmung / Ganzheitliche Betrachtung**	**92**

1 Einleitung

1.1 Teil 1

1.1.1 Motivation



Anwendungsentwicklerinnen und Anwendungsentwickler scheuen das Tageslicht und die Menschen. Sie sitzen im Keller, spielen ständig LAN-Partys und haben im Unternehmen nur drei Aufgaben: 1. Aufwandschätzung, 2. Programmierung und 3. Fehlerbeseitigung.

Wirklich?

Sehr geehrte Leserin, sehr geehrter Leser,

die IT ist Segen und gleichzeitig Fluch. Richtig angewendet, kann Sie eine signifikante Qualitätsverbesserung bei gleichzeitiger Kostensenkung bewirken. Falsch eingesetzt, kann Sie genau das Gegenteil erreichen. Ein ideal aufeinander abgestimmtes Team, bestehend aus einem zukunftsorientierten + aufgeschlossenen Kunden und erfahrenden + innovativen Prozessberatern, Anwendungsberatern, Anwendungsbetreuern,

Anwendungsentwicklern und Projektmanagern, ist Voraussetzung und Garant für den positiven Fall.

Die Anwendungsentwicklung ist essentieller Bestandteil dieses Teams!

Doch wo Licht ist, gibt es auch Schatten: Richtig eingesetzt, kann die Anwendungsentwicklung für die notwendige Flexibilität bei der Anpassung einer IT-Lösung sorgen bzw. die Anwendungsberatung und -betreuung intensiv unterstützen und entlasten. Falsch "genutzt", kann sie die IT-Lösung destabilisieren, deflexibilisieren und verkomplizieren. Darüber hinaus kann sie eine dauerhafte Abhängigkeit schaffen, da Support und Weiterentwicklung nur durch die Anwendungsentwicklung möglich sind. Außerdem besteht die Gefahr eines erhöhten Sicherheitsrisikos durch unerlaubte Datenzugriffe, Viren und Trojaner. Die Lösung: Eine bewusst gelebte Serviceorientierung + Produktisierung in der Anwendungsentwicklung kann die nahezu vollständige Eliminierung dieser negativen Aspekte bewirken.

Dieses Ziel ist nur einen Katzensprung entfernt, denn: Die Anwendungsentwicklung lebt in der Regel bereits unbewusst ein vollständig ausgeprägtes IT Service Management und eine damit verbundene Serviceorientierung + Produktisierung!

Hoher Nutzen für die Anwendungsberatung und Anwendungsbetreuung

Nutzen für Beratung / Betreuung:

- Veränderung und Erweiterung des Standards
- Entwicklung von Zusatz- und Individuallösungen
- Suche nach (geheimen) Customizingtabellen, Reports und Funktionen
- Analyse von Stammdaten-, Customizing- und Softwarefehlern
- Allgemeine technische Unterstützung (z.B. Architektur)

Gefahren für Beratung / Betreuung:

- Hohe Fehlerrate, Niedriger Reifegrad, Geringe Flexibilität
- Erhöhte Abhängigkeit (Kein Support durch SAP)
- Sicherheitsrisiko (Datenzugriffe, Viren, Trojaner)

Keine Gefahr bei bewusst gelebter Serviceorientierung und Produktisierung

In diesem Handbuch werden durch Verbindung von Theorie und Praxis diese und weitere Thesen plausibel und anschaulich belegt:

- Die Anwendungsentwicklung ist ein für Anwendungsberatung und -betreuung essentieller IT-Dienstleister.
- Die Anwendungsentwicklung kann viel mehr leisten, als meist wahrgenommen wird.
- Das IT Service Management ist nicht nur Theorie, sondern unbewusst gelebte Praxis der Anwendungsentwicklung.
- Das in der Anwendungsentwicklung gelebte Custom Code Lifecycle Management ist eine weitestgehend unbekannte aber legitime Spezialisierung / Konkretisierung des IT Service Managements.
- Die bewusste Serviceorientierung + Produktisierung ist eine große Chance zur Qualitätssteigerung bei gleichzeitiger Kostensenkung.

Dieses Handbuch hilft Ihnen bei der Verifizierung dieser These und beim damit verbundenen Übergang zu einer bewusst gelebte Serviceorientierung (+ Produktisierung). Es soll Sie sensibilisieren und wichtige Denkanstöße liefern. Dank dieses Handbuchs (und weiterer Literaturquellen) sind Sie in der Lage, die Qualität Ihrer Anwendungsentwicklung zu verbessern und gleichzeitig deren Kosten signifikant zu senken…

Ich wünsche Ihnen viel Freude beim Lesen und Handeln.

Mit freundlichem Gruß

Peter Weigel

1.1.2 Versionshistorie

Version	Autor	Datum	Bemerkungen
1	Peter Weigel	14.07.2014	Erste Version

1.1.3 Ausgangssituation

Es existieren im Unternehmen SAP-/ABAP-Anwendungsentwickler als Teil der Anwendungsberatungsabteilungen und Anwendungsbetreuungsabteilungen oder als eigenständige Anwendungsentwicklungsabteilung. Es besteht der Wunsch nach signifikanten Effizienz-, Effektivitäts- und Qualitätssteigerungen.

1.1.4 Zielsetzung

Dieses Handbuch richtet sich im Allgemeinen an alle Anwendungsentwickler im Umfeld der SAP Business Suite und im Speziellen an alle Leiter einer SAP-Anwendungsentwicklungsabteilung. Darüber hinaus ist dieses Handbuch mit geringfügigen Abstrichen auch für Anwendungsentwicklungen ohne SAP-Bezug geeignet.

Das Handbuch dient als Hilfsmittel für Aufbau oder Weiterentwicklung einer Anwendungsentwicklungsabteilung und als Mittel zur Rollenschärfung / Motivation für Anwendungsentwicklerinnen und Anwendungsentwickler mit dem Ziel der signifikanten Effizienz-, Effektivitäts- und Qualitätssteigerung.

Das Handbuch ist als Ergänzung der SAP Publikationen im Umfeld Application Lifecycle Management, RunSAP und SAP Solution Manager anzusehen und stellt gleichzeitig eine Complementary-Publikation des ITIL-Frameworks dar.

Außerdem dient dieses Handbuch dazu, die Rolle des von SAP identifizierten bzw. kommunizierten E2E-/ALM-Prozesses "Custom Code Management" in das "globalgalaktisch" richtige Licht zu rücken: Jedes IT-Unternehmen lebt bewusst oder unbewusst ein vollumfängliches IT Service Management (ITSM). Ein Anwendungsberatungs-, Anwendungsbetreuungs- oder Softwarehaus lebt die Spezialisierung des Application Lifecycle Managements (ALM). Das ALM ist dabei als legitime Konkretisierung des ITSM für Softwarehäuser mit dem Fokus auf Implementierung und Betrieb von Softwarelösungen und ist somit nicht als Konkurrenz mit teilweisen Überschneidungen zu verstehen. Im Umfeld der SAP Business Suite gelten die Spezialisierungen der End-To-End Solution Implementation and Operation Standards (E2E). SAP stellt in diesem Kontext wichtige, bewährte und leistungsstarke Best Practices, Hilfsmittel und Werkzeuge (z.B. im SAP Solution Manager) zur Verfügung. Bei genauerer Betrachtung der von SAP kommunizierten E2E-/ALM-Prozesse und -Tools lässt sich ein eindeutiges Mapping zu den IT Service Management Prozess herstellen. Mit einer Ausnahme: Das Custom Code Management (CCM). Ist das CCM damit ein bisher unentdeckter ITSM-Prozess? Nein! Denn dieses Handbuch belegt, dass es sich bei dem Custom Code (Lifecycle) Management um eine weitere Spezialisierung des Application Lifecycle Managements handelt, welche für alle IT-Dienstleister relevant ist, die eine Standardsoftware modifizieren und erweitern. Sämtliche Publikationen zum Application Lifecycle Management, zum Custom Code Lifecycle Management, zum Template Management (als Spezialisierung des Custom Code Lifecycle Managements für Template-Rollout- bzw. Master-Lokalisierungs-Szenarien), zu den End-To-End Solution Implementation and Operation Standards oder zu dem SAP Solution Manager können in diesem Sinne als offizielle ITIL-Ergänzungs-Publikationen (ITIL Complementary Guidances) angesehen werden.

1.1.5 Abgrenzung

Dieses Handbuch ist als Ergänzung zu den End-To-End Solution Implementation and Operation Standards "Custom Code Management" und "Upgrade Management", zu den

ITIL-Publikationen zum Thema "IT Service Management" bzw. "Application Lifecycle Management" und zu den zahlreichen Publikationen zum Thema "Software Engineering" anzusehen. Es ist kein Ersatz.

Die Kenntnisse über die Details der beschriebenen Aktivitäten, inkl. zur Anwendung der genannten Werkzeuge, werden vorausgesetzt und somit nicht ausführlich besprochen. Schwerpunkt dieses Dokumentes ist ein allgemeiner Überblick.

1.2 Teil 2

1.2.1 Literatur

Dieses Handbuch richtet sich im Allgemeinen an alle Anwendungsentwickler im Umfeld der SAP Business Suite und im Speziellen an alle Leiter einer SAP-Anwendungs-entwicklungsabteilung.

Neben diesem Handbuch sei noch auf die beiden Bücher "SAP Solution Manager (3. Auflage)" (SAP PRESS, ISBN 978-3-8362-1737-8) und "Run SAP - Das umfassende Handbuch" (SAP PRESS, ISBN 978-3-8362-1432-2) sowie die SAP-Schulung "E2E500 - Custom Code Management" verwiesen. Der Fokus der beiden Bücher und der genannten Schulungen liegt auf den unterstützenden SAP Solution Manager Werkzeugen und der Eingliederung in die SAP End-To-End Solution Operation Standards.

Für mehr Theorie lohnt sich ein Blick in die zahlreichen Nachschlagewerke zum Thema "IT Service Management / IT Infrastructure Library" und "Softwareentwicklung / Produkt-lebenszyklus / Service orientierte Architektur / Software & Systems Engineering".

Für mehr "Input" zur ABAP-Anwendungsentwicklung sei auf die zahlreichen Literaturquellen zum Thema (in der Regel von SAP PRESS) verwiesen.

1.2.2 Rechtliche Hinweise

http://www.hybrid-eichhörnchen.de/rechtliche-hinweise/

1.2.3 Kontakt

Sehr geehrte Damen und Herren.

Ich möchte gern mein Wissen und meine Erfahrungen mit Ihnen teilen. Hierzu biete ich Ihnen ausgewählte Leitfäden und Dokumentationen zur kostenfreien Nutzung an. Das vorliegende Dokument ist Bestandteil dieses Kompendiums. Es ist auf Grundlage meiner in Beratungsprojekten gewonnenen Erkenntnisse und Erfahrungen entstanden und wird kontinuierlich weiterentwickelt.

Ich hoffe, dieses Dokument wird Ihnen helfen, Ihre IT wertschöpfend auszuprägen und einzusetzen. Wenn Sie hierzu Fragen haben, Unterstützung wünschen oder zusätzliche Hilfe benötigen, dann stehe ich Ihnen gern zur Seite!

Hat Ihnen das Dokument gefallen und geholfen? Dann teilen Sie mir dies bitte mit, damit ich dieses Angebot effektiv pflegen und ausbauen kann. Wenn Sie Fehler entdecken, konkrete Verbesserungsvorschläge haben und aktiv an der Aktualisierung und Weiterentwicklung des Dokumentes mitwirken möchten, dann nehmen Sie bitte Kontakt zu mir auf.

Ich freue mich auf Ihr Feedback!

Peter Weigel

Peter Weigel
Hyazinthenstraße 6
D-06122 Halle an der Saale
Telefon: +49 170 5337567
E-Mail: peter.weigel@hybrid-eichhörnchen.de
Web: www.hybrid-eichhörnchen.de

1.2.4 Download

Die aktuelle Version dieses Dokumentes können Sie stets kostenfrei und ungekürzt unter www.hybrid-eichhörnchen.de herunterladen.

2 Anwendungsentwicklung

Folgende Abschnitte dienen als Einstimmung / Motivation in das Thema "Anwendungsentwicklung" und besitzen teilweise nur einen indirekten Bezug zu dem Kernthema dieses Handbuchs.

2.1 Interview mit einem fiktiven Entwicklungsleiter

Von der Programmierung über die Entwicklungsberatung zum Custom Code Lifecycle Management.

Herr Entwicklungsleiter, seit wann sind Sie Leiter der Anwendungsentwicklungsabteilung?

Seit deren Gründung vor vielen Jahren. Damals wurde entschieden die Anwendungsentwicklungskompetenzen aus den Anwendungsberatungs- und Anwendungsbetreuungsabteilungen auszulagern und in eine eigenständige Abteilung zu bündeln.

Damals eine richtige Entscheidung?

Auf jeden Fall! Anwendungsentwickler lieben die Herausforderung im Umgang mit Algorithmen und Techniken, sie sind quasi Forscher, Erfinder und Zauberer in Personalunion und unterscheiden sich damit grundlegend von Projektmanagern, Anwendungsberatern oder Anwendungsbetreuern für die eher die Kommunikation mit Kunden bzw. Projektteammitgliedern im Vordergrund steht. Zudem sind die Anforderungen an die Anwendungsentwicklung enorm. Die Theorien, Techniken und Methoden der Anwendungsentwicklung sind leistungsfähig und flexibel aber auch sehr komplex. Um die Möglichkeiten voll ausschöpfen zu können, muss man zwangsläufig Experte / Spezialist sein...und das geht nur im Team.

Welche Rolle hat die Anwendungsentwicklung in Ihrem Unternehmen heute?

Wir sind ein gefragter bzw. begehrter IT-Service-Provider und bieten den Fachabteilungen unseres Unternehmens den IT Service "Programmierung, Entwicklungsberatung und Kundenentwicklungsmanagement" in Form von konkreten Anwendungsentwicklungsleistungen und gut ausgebildeten Anwendungsentwicklerressourcen an. Wir leben hierbei ein vollumfängliches ITIL-konformes IT-Service-Management in der praktischen Spezialisierung des Custom-Code-Lifecycle-Managements, das wiederum eine Unterart des Application-Lifecycle-Managements darstellt.

ITIL? Da denkt man eher an Incident Management & Co!?

Wenn eine Anwendungsentwicklerressource krankheitsbedingt ausfällt und damit den Projektfortschritt gefährdet, ist das im Sinne von ITIL eine ‚Störung' / ein ‚Incident' unseres

Services. Wenn ein Entwicklungsberater eine Schulung besucht, dann erfährt unser Service eine Verbesserung und damit eine ‚Änderung' / ein ‚Change'. Und wenn mehrere Programmierer neu eingestellt werden, dann passiert das bei uns im Sinne des ‚Release Managements' ITIL-konform zum Monatsanfang.

Wir bilden quasi das Fundament der Servicearchitektur unseres Unternehmens, denn wir betreiben Services (z.B. Programmanalyse, Fehlerbeseitigung, Konzeption, Programmierung, Upgrade und Qualitätssicherung), generieren Services (neue Add-on Services) und unterstützen die Beratung und Betreuung bei ihren Consulting und Support Services...eine sehr anspruchsvolle Leistung.

Und was genau "leistet" ein Anwendungsentwickler, wenn er nicht gerade an einer LAN-Party teilnimmt?

Die LAN-Partys gibt es wirklich und da nehmen nicht nur Anwendungsentwickler teil...Die wahrnehmungsstärkste Tätigkeit ist natürlich das Programmieren in verschiedensten Modulen (z.B. SAP HCM) oder Techniken (z.B. SAP Workflow). Aber in Wahrheit ist das Tätigkeitspektrum viel fassettenreicher und erstreckt sich von der Erstellung oder Beurteilung von Konzepten über die Programmierung (inkl. Coaching, Entwicklungskoordination, Qualitätssicherung), Testdurchführung und -automatisierung, Produktivsetzung, Programmanalyse und Fehlerbeseitigung bis hin zu Weiterentwicklung, Rückbau, Rückführung in den Standard und Pflege von Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation.

Scheint so, als wären Sie für gegenwärtige und zukünftige Aufgaben bestens aufgestellt?

Das ist richtig: Wir haben ein konkretes Ziel vor Augen und sind auf dem richtigen Weg. Aber: Obwohl wir schon viel erreicht haben, ist noch einiges zu tun. Ein Beispiel: Aktuell ist es leider noch so, dass interne und externe Anwendungsentwickler unterschiedlich verrechnet werden. Das ist für die Beratungsprojekte nicht kalkulierbar / planbar. Diese Situation hatten wir früher bei unseren Dienstfahrzeugen übrigens auch: Fahrzeuge aus unserem Fuhrpark waren kostenlos und Mietautos waren teuer. Seitdem das umgestellt wurde, also Fuhrpark und Mietfahrzeuge gleichgestellt und die Buchungen zentralisiert sind, haben wir endlich vollständige Transparenz und Planbarkeit, sowohl für die Projekte als auch für das Fuhrparkmanagement. Ein anderes Beispiel: Bisher waren wir ausschließlich für unsere internen Beratungs- und Betreuungsabteilungen tätig. Zukünftig planen wir externen IT-Dienstleistern (und EDV-Abteilungen) standardisierte + lukrative Entwicklungsberatungsleistungen anzubieten mit dem Ziel, die Türen bei neuen Kunden zu öffnen und gleichzeitig einen wesentlichen Beitrag zur Umsatz- bzw. Gewinnerwirtschaftung unseres Unternehmens zu leisten. Soviel zu zwei konkreten Aktivitäten, die wir gerade angehen...uns wird also nicht langweilig...zumal wir intern viele weitere Ideen diskutieren...

Jetzt machen Sie uns aber neugierig. Was haben Sie denn noch so vor?

Wenn Sie mich so Fragen...Wir haben noch viele weitere Ideen, die zeitnah zu untersetzen und anzugehen sind: Ersetzen der Auslastungsvorgaben unserer Abteilung

durch Gewinnzielvorgaben, Eigenverantwortlicher Knowhow- und Personalaufbau, Eigenverantwortlicher + zentralisierter Entwicklerressourceneinkauf, Vereinbaren von verbindlichen Operational Level Agreements mit unseren Beratungs- und Betreuungsabteilungen, Durchsetzen eines verbindlichen Ressourcencommitments inkl. Einführung von Stornogebühren, Projektunabhängige Übernahme der Verantwortung für die Lösungsqualität inkl. Etablierung von verbindlichen Quality Gates, Technologische Spezialisierung / Fokussierung der Anwendungsentwickler und Ausbildung von Technologieverantwortlichen, Aufbau eines kostenfreien Coaching- und Qualitätssicherungsservices, Gründung eines Competence Teams / Centers "ABAP Anwendungsentwicklung" zur besseren Integration der entwickelnden Berater, Aufbau einer Add-on Suite und Ersetzen sämtlicher kundenspezifischer Entwicklungslösungen durch wiederverwendbare Add-ons, Einführung eines ganzheitlichen IT Service Managements in der Anwendungsberatung und -entwicklung, ...

Das klingt spannend. Herr Entwicklungsleiter, vielen Dank für das Gespräch und weiterhin bzw. für die anstehenden Aktivitäten viel Erfolg!

2.2 Theorie und Praxis eines fiktiven Entwicklers

Theorem:

- Nur ein guter Theoretiker ist ein guter Anwendungsentwickler.
- Nur ein guter Praktiker ist ein guter Anwendungsentwickler.

Beweis (Teil 1):

- Die Haupttätigkeiten eines Anwendungsentwicklers bzw. -beraters während
- der Grobkonzeption, fachlichen Konzeption, technischen Konzeption,
- Implementierung und Dokumentation sind theoretischer Natur:
- Erkenne und identifiziere ein kompliziertes Problem.
- Analysiere und vereinfache das Problem.
- Finde die Ursache des Problems.
- Entwickle eine hochinnovative Lösung.
- Vereinfache und verallgemeinere die Lösung.
- Beweise die Vollständigkeit und Korrektheit der Lösung.
- Realisiere / Implementiere die Lösung in absolut höchster Qualität.
- Dokumentiere die Lösung verständlich für Anwender, Berater und Anwendungsentwickler.

Beweis (Teil 2):

- Die Ergebnisse der Anwendungsentwicklung bzw. -beratung
- müssen zahlreichen praktischen Ansprüchen genügen:
- Korrektheit, Vollständigkeit, Sicherheit, Reife,
- Fehlertoleranz, Wiederherstellbarkeit,
- Verständlichkeit, Erlernbarkeit,
- Bedienbarkeit, Ergonomie,
- Zeitverhalten, Verbrauchsverhalten,
- Unabhängigkeit, Komplexität, Genauigkeit,
- Analysierbarkeit, Modifizierbarkeit, Erweiterbarkeit,
- Stabilität, Prüfbarkeit, Installierbarkeit, Austauschbarkeit, ...

q.e.d.

2.3 Mission einer fiktiven Anwendungsentwicklungsabteilung

Sie benötigen Hilfe? Wir können helfen! Gemeinsam verbessern wir die Welt.

Ihr Bedarf: Sie hätten gern eine Beschleunigung durch teilweise oder vollständige, 100% transparente Automatisierung Ihrer Geschäftsprozesse? Sie erhoffen sich eine Vermeidung von Irrwegen und Missbräuchen durch konsequente Führung, Validierung, Unterstützung und passgenaue Berechtigungsvergabe? Sie wünschen eine maximale Vereinfachung durch praxistaugliche Bedienbarkeit, größtmögliche Komplexitätsreduktion und bestmögliche Effizienzsteigerung? Kurz und Knapp: Sie träumen von einer höchstmöglichen Qualitätsverbesserung mit gleichzeitiger Kostensenkung durch intelligenten Einsatz der IT?

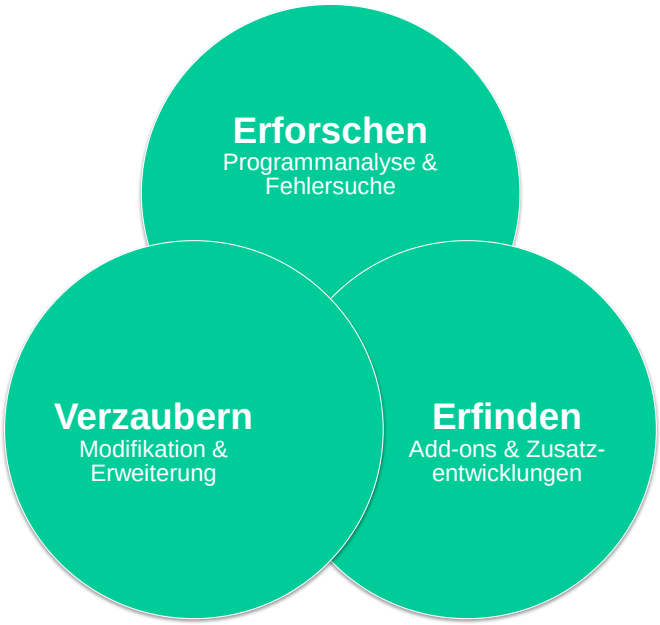
Dann kommen Sie zu uns.
Wir machen Ihre Träume wahr!

Gemeinsame Überzeugung: Wir leben in einer schnelllebigen, komplex-innovativen Welt. Einer Welt, in der sämtliche Geschäftsprozesse nur noch mittels der Informationstechnologie beherrschbar sind und / oder signifikant verbessert werden können. Die IT hat sich damit schleichend von einem kleinen Hilfsmittel zu einem wichtigen, unabdingbaren Unternehmenserfolgsfaktor entwickelt. Wir sind überzeugt, dass ein echter Mehrwert hinsichtlich Qualität & Effizienz und damit einhergehend ein maximaler Beitrag zu Ihrem Unternehmenserfolg, nur durch eine vollständige, konsistente und optimal aufeinander abgestimmte Business- & IT-Lösung möglich ist.

Sie stimmen dem uneingeschränkt zu?
Dann leben Sie diese Überzeugung!
Wir unterstützen Sie dabei.

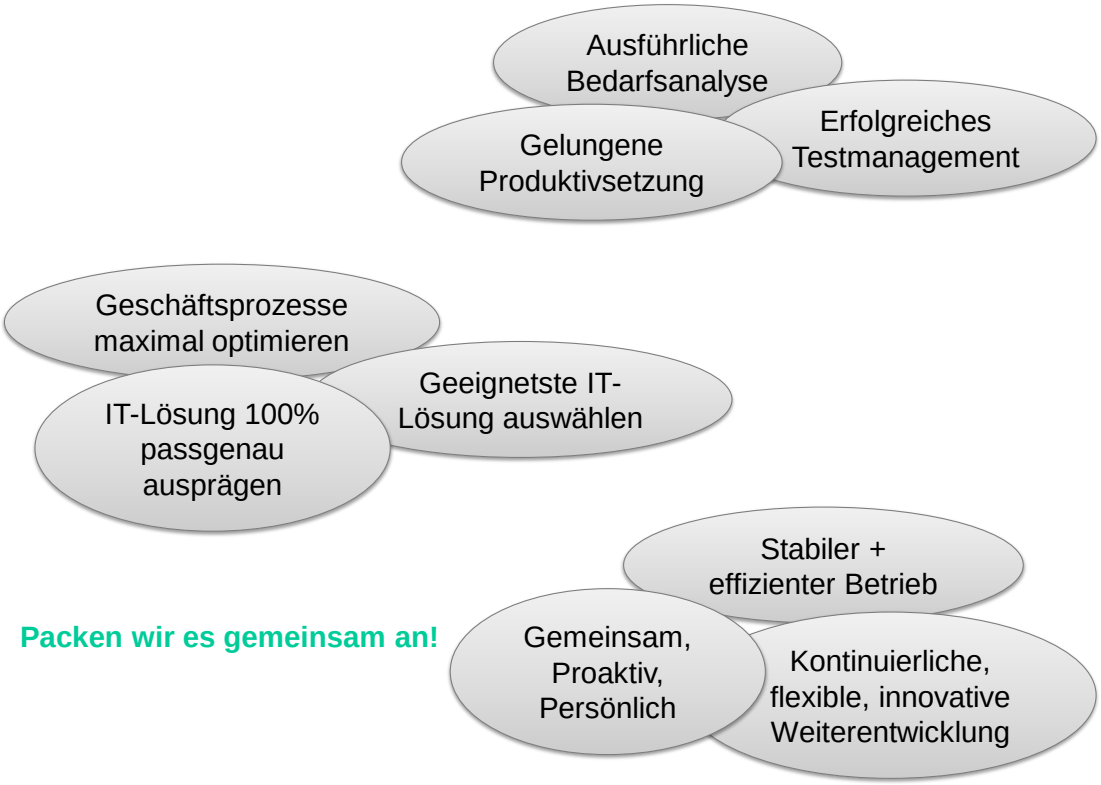
Unsere Stärke: Der Einsatz einer in der Praxis bewährten Standardsoftwarelösung wie der SAP-Business-Suite ist aus Gründen der Stabilität, Flexibilität, Vollständigkeit, Konsistenz, Effizienz, Effektivität, Transparenz, Qualität und Kosten in den meisten Fällen einer Individualentwicklung vorzuziehen. Diese Standardlösung vereint zahlreiche erprobte Best-Practices und lässt sich darüber hinaus unternehmensspezifisch anpassen. In einigen Fällen stößt die Konfiguration der SAP-Standardlösung jedoch an ihre Grenzen, da die Anpassung durch Konfiguration nur in dem Maße möglich ist, wie dies vom Hersteller vorgesehen wurde. Und hier kommt die Anwendungsentwicklung ins Spiel: Neben dem Erfinden komplett eigenständiger Zusatzentwicklungen, also Funktionalitäten, die vom Hersteller aus verschiedenen Gründen überhaupt nicht angedacht waren, sind Anwendungsentwickler in der Lage, den Standard beliebig und ohne Einschränkungen zu verzaubern und hierbei durch Modifikation und Erweiterung echte, einmalige, noch nie dagewesene Innovationen zu implementieren und nahtlos zu integrieren. Darüber hinaus können Anwendungsentwickler, durch ihr Wissen und ihre Fähigkeiten in der Programmanalyse, die Anwendungsberatung und -betreuung bei der Forschung nach (geheimen) Customizingtabellen, Stammdateninkonsistenzen, Konfigurations- und Softwarefehlern tatkräftig unterstützen.

Wir Erfinder, Forscher und Zauberer lieben Herausforderungen!
Sie auch?



Unsere Mission: Wir verstehen unsere Mission darin, gemeinsam mit Ihnen und unseren Beratungs- und Betreuungskollegen Ihre Geschäftsprozesse maximal zu optimieren, die geeignetste Standard-IT-Lösung auszuwählen und diese gemäß vorangehender ausführlicher Bedarfsanalyse mittels Konfiguration und Programmierung 100% passend für Ihr Unternehmen auszuprägen. Die Welt steht nach erfolgreichem Test und gelungener Produktivsetzung jedoch nicht still: Auch ein anschließender stabiler + effizienter Betrieb und die kontinuierliche + flexible + innovative Weiterentwicklung Ihrer-Unserer Business- & IT-Lösung ist uns eine Herzensangelegenheit!

Packen wir es gemeinsam an!
Ihre Anwendungsentwicklung.

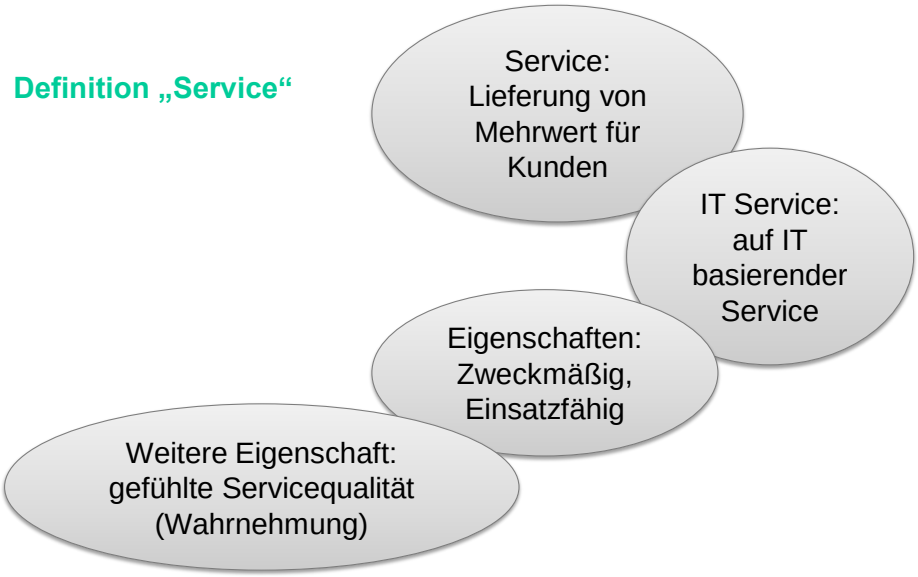


3 IT-Dienstleistungen (IT Services)

3.1 IT-Dienstleistung und Wertdefinition (IT Service Definition)

Was ist eine Dienstleistung? Laut der IT Infrastructure Library (ITIL) ist eine Dienstleistung (engl.: Service) eine Möglichkeit zur Lieferung eines Mehrwertes für Kunden, mit dem diese angestrebte Ergebnisse einfacher erreichen können ohne die Verantwortung für bestimmte Kosten und Risiken selbst zu tragen. Eine IT-Dienstleistung (engl.: IT Service) ist eine auf dem Einsatz der Informationstechnologie basierende Dienstleistung und somit eine Spezialisierung des Dienstleistungsbegriffs.

Doch was bedeutet “Mehrwert für Kunden”? Was macht den Wert einer Dienstleistung aus? Laut der Wertdefinition nach ITIL, wird der Wert einer Dienstleistung durch die drei Eigenschaften "Zweckmäßigkeit", "Einsatzfähigkeit" und "Wahrnehmung" bestimmt. Ein Service ist zweckmäßig (engl.: Utility), wenn seine Funktionalität die Verbesserung der Leistungsfähigkeit oder die Beseitigung von Beschränkungen bewirkt. Ein Service ist einsatzfähig (engl.: Warranty), wenn Zusagen / Garantien der vereinbarten Anforderungen hinsichtlich Verfügbarkeit, Kapazität, Kontinuität (= Aufrechterhaltung + Wiederherstellung) und Sicherheit existieren und gesichert sind. Zusätzlich zu diesen "harten" Faktoren wird laut ITIL die Wahrnehmung bzw. die "gefühlte Servicequalität" in einem teilweise erheblichen Maß durch "weiche" Faktoren wie "zurückliegende Erfahrungen des Kunden", "Vergleich mit Wettbewerbern", "Image + Position des Anbieters" etc. indirekt beeinflusst.



Gemäß dieser Definitionen handelt es sich bei "Entwicklungsberatung und Programmierung" als Sammelbegriff für alle Tätigkeiten der Anwendungsentwicklung zweifelsfrei um eine Dienstleistung. Sie ist zweckmäßig und bei Bereitstellung ausreichender, befähigter Entwicklerressourcen auch einsatzfähig.

Doch handelt es sich bei "Entwicklungsberatung und Programmierung" um die einzige Dienstleistung der Anwendungsentwicklungsabteilung? Denkbar wären ja auch die Dienstleistungen "CRM-Anwendungsentwicklung", "ERP-Anwendungsentwicklung" und "HCM-Anwendungsentwicklung", also eine Zerlegung gemäß der verschiedenen SAP-Module. Möglich wären aber auch "Workflow-Anwendungsentwicklung", "Smartform-Anwendungsentwicklung" und "Webdynpro-Anwendungsentwicklung", also eine Betrachtung der Technologien. Auch eine Fokussierung auf konkrete Tätigkeiten wie "Technische Konzeption", "Programmierung" und "Qualitätssicherung" wäre möglich.

Die Antwort lautet hier: Ja und Nein. Ja, es handelt sich um mögliche Unterdienstleistungen (engl.: Subservices) - Die Aufstellung einer Servicehierarchie in Form eines Servicebaumes ist hierbei allerdings nicht ganz so einfach realisierbar, da die Unterdienstleistungen Überschneidungen aufweisen. So ist die Tätigkeit "Technische Konzeption für eine Smartform-Anwendungsentwicklung im SAP CRM" Bestandteil der drei verschiedenen Unterdienstleistungen "Technische Konzeption", "Smartform-Anwendungsentwicklung" und "CRM-Anwendungsentwicklung".

Zum Glück, sonst würden wir uns mit diesem Thema noch sehr lange und ausführlich beschäftigen müssen, lautet die Antwort meist jedoch "Nein". Denn wie wir sehen, kann

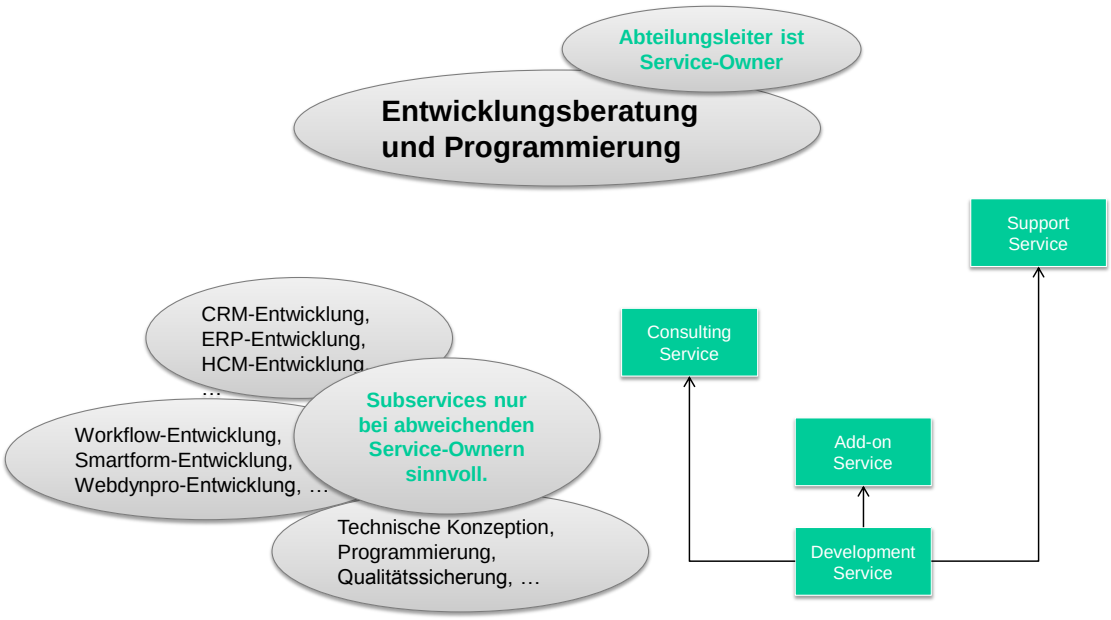
eine Dienstleistung theoretisch nahezu immer auf verschiedenste Weise in Unterdienstleistungen zerlegt werden. Wirklich Sinn macht diese Zerlegung aber nur, wenn verschiedene Dienstleistungsverantwortliche/-eigentümer (engl.: Service Owner) und verschiedene Dienstleistungsvereinbarungen (engl.: Service Level Agreements) existieren. In der Regel ist jedoch der Leiter der (einzigen im Unternehmen existierenden) Anwendungsentwicklungsabteilung der Verantwortliche / Eigentümer der Dienstleistung(en).

Gibt es aber vielleicht "Überdienstleistungen"? Dienstleistungen können in Dienstleistungen zerlegt werden. Analog lassen sich Dienstleistungen zu Dienstleistungen zusammensetzen. Außerdem können Dienstleistungen durch andere Dienstleistungen konsumiert / genutzt werden. Mit diesem Thema werden wir uns im nächsten Abschnitt noch einmal beschäftigen. An dieser Stelle ist es jedoch wichtig zu erkennen, dass die Dienstleistung "Entwicklungsberatung und Programmierung" zum einen von der Dienstleistung "Anwendungsberatung" und zum anderen von der Dienstleistung "Anwendungsbetreuung" konsumiert wird. Diese Dienstleistungen liegen logischerweise nicht in der Verantwortung der Entwicklungs-, sondern der Beratungs- bzw. Betreuungsabteilung und sind damit keine "Überdienstleistungen" der Anwendungsentwicklungsabteilung.

Dienstleistungen bewirken gemäß ihrer Zweckmäßigkeitseigenschaft eine Verbesserung der Leistungsfähigkeit oder die Beseitigung von Beschränkungen. Das sind Ziele, die durch konkrete Aktivitäten erreicht werden sollen. Diese Aktivitäten haben konkrete Ergebnisse zur Folge, anhand derer sich die Zielerreichung verifizieren lässt. Ein mögliches Ergebnis der Dienstleistung "Entwicklungsberatung und Programmierung" ist ein konkretes ABAP Add-on, das im Rahmen einer konkreten Entwicklungstätigkeit erstellt wird. Dieses Add-on ist zweckmäßig + einsatzfähig und damit selbst wieder eine Dienstleistung. Die Dienstleistung "Entwicklungsberatung und Programmierung" ist also in der Lage neue Dienstleistungen zu erzeugen und zu betreiben.

Aber: ABAP Add-ons sind keine Dienstleistungen der Anwendungsentwicklung. Anwendungsentwickler sind zuständig, aber nicht verantwortlich. Identifizierung potentieller und Vermarktung existierender Add-ons erfolgen ausnahmslos durch die Anwendungsberatung. Damit ist die Anwendungsberatung Eigentümer konkreter Add-on-Dienstleistungen. "Add-on Services" sind damit ebenfalls keine "Überdienstleistungen" der Anwendungsentwicklungsabteilung.

Fazit: Die Anwendungsentwicklung besitzt nur die Dienstleistung "Entwicklungsberatung und Programmierung" und nur diese! Einzige Ausnahme: Ein Unternehmen besitzt tatsächlich mehrere Entwicklungsabteilungen - z.B. Beratung vs. Betreuung oder ABAP vs. JAVA.

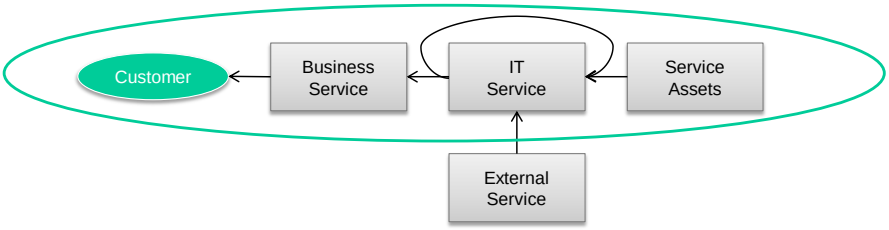

Abteilungsleiter ist Service-Owner
Entwicklungsberatung und Programmierung
Support Service
CRM-Entwicklung, ERP-Entwicklung, HCM-Entwicklung …
Consulting Service
Subservices nur bei abweichenden Service-Ownern sinnvoll.
Workflow-Entwicklung, Smartform-Entwicklung, Webdynpro-Entwicklung, …
Add-on Service
Technische Konzeption, Programmierung, Qualitätssicherung, …
Development Service

3.2 Komposition von IT-Dienstleistungen (IT Service Composition)

Laut ITIL ist ein Business Service ein für den Endkunden sichtbarer Service. Ein IT Service basiert auf dem Einsatz der Informationstechnologie. Ein externer Service ist ein von einem externen Service Provider bereitgestellter Service. Und: Ein Subservice ist ein von einem anderen Service konsumierter Service.

Die Klassifikationen "Business", "IT", "External" und "Sub" sind kombinierbar. So kann zum Beispiel ein Business Service auch gleichzeitig ein externer IT-Service sein und als Subservice von einem anderen Service konsumiert werden. Üblich ist dies jedoch nicht. Denn meistens stehen für Kunden sichtbare Geschäftsdienstleistungen (Business Services) ausschließlich für Endkunden zur Verfügung und basieren auf vor dem Kunden verborgenden IT-Dienstleistungen, welche wiederum auf andere interne oder externe IT-Dienstleistungen zurückgreifen können oder bei genauerer Betrachtung in zahlreiche Dienstleistungsbausteine (Service Assets) zerfallen werden. Damit entsteht eine komplexe Servicearchitektur.

- **Business Service**: Ein Business Service ist ein für den Endkunden sichtbarer Service.
- **IT Service**: Ein IT Service basiert auf dem Einsatz der Informationstechnologie.
- **External Service**: Ein externer Service ist ein von einem externen Service Provider bereitgestellter Service.
- **Subservice**: Ein Subservice ist ein von einem anderen Service konsumierter Service.
- Die Klassifikationen „Business“, „IT“, „External“ und „Sub“ sind kombinierbar.



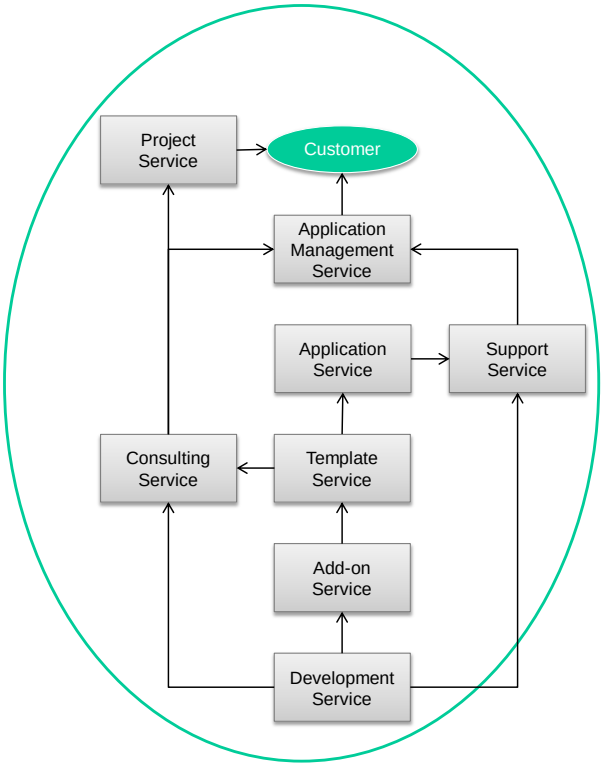
Die Entwicklungsdienstleistung (Development Service) ist Teil dieser Servicearchitektur:

- Die Eigenschaften "Zweckmäßigkeit" und "Einsatzfähigkeit" sind durch Service / Operational Level Agreements abgesichert.
- Die konsumierenden Services reichern den "Development Service" um weitere Funktionalitäten an und erhöhen damit den Gesamtmehrwert.
- Die konsumierenden Services lagern die Technologieverantwortung an den "Development Service" aus und reduzieren damit Komplexität und Risiko.

Der Development Service dient als Basis für konkrete Add-on Services (kleine Softwarebausteine). Die Add-on Services wiederum fließen in konkrete Template Services (komplette Softwarelösungen bestehend aus Entwicklungen und Konfigurationen) ein, die wiederum Bestandteil von Application Services (Betrieb von kompletten Softwarelösungen im ASP- bzw. SaaS- bzw. Cloud-Szenario) sein können. Template Services und Development Services werden von Consulting Services (generische Beratungsleistungen) konsumiert. Für den Kunden sichtbar sind jedoch lediglich konkrete Project Services (temporäre Anwendungsberatungs- bzw. Projekteinsätze) oder Application Management Services (dauerhafte Anwendungsbetreuung und kontinuierliche Weiterentwicklung). Application Services und Development Services werden von Support Services (generische Betreuungsleistungen) genutzt. Der Support Service ist wiederum Bestandteil des für den Kunden sichtbaren Application Management Services.

Der "Development Service" wird, wie bereits erwähnt, vom "Consulting Service" als Subservice konsumiert und hierbei nahtlos in den sich über mehrere Lebenszyklusphasen und Architekturschichten erstreckenden Wertlieferprozess mit Schwerpunkt "Technische Konzeption" und "Implementierung" integriert. Add-ons und Templates reduzieren die direkte Konsumierung des "Development Services" und beschleunigen die Phasen der "Technischen Konzeption" und "Implementierung" bei gleichzeitiger Qualitätssteigerung und Kostenreduktion. Der "Consulting Service" ist ein generischer Service für den Vertrieb des IT-Dienstleisters (Die Beratung garantiert dem Vertrieb die Realisierbarkeit von Projekteinsätzen. Der Vertrieb garantiert der Beratung die Beschaffung von Kunden.) und für die Anwendungsbetreuung (Die Beratung garantiert der Anwendungsbetreuung die Realisierbarkeit von Unterstützungs- und Weiterentwicklungseinsätzen. Die Anwendungsbetreuung garantiert der Anwendungsberatung die Einbeziehung in entsprechende Aktivitäten bzw. die komplette Übertragung großer Weiterentwicklungsvorhaben/-projekte.). Der "Consulting Service" wird zum "Project Service" bzw. zum Bestandteil des "Application Management Service" für den Kunden bei konkreter Besetzung der Ressourcen, Aufgaben, Budgets, Termine, … und der damit einhergehenden Konkretisierung der Service Level Agreements. In analoger Weise ist die Betreuungsdienstleistung (Support Service) für die Softwarebetriebs- und -optimierungsphasen zuständig. Der "Development Service" wird hier ebenfalls als Subservice mit Schwerpunkt "Fehlerbeseitigung und Weiterentwicklung" konsumiert. Der Einsatz von ASP-Lösungen reduziert die direkte Konsumierung des "Development Services" und senkt die Kosten für den Betrieb. ...

- Die konsumierenden Services reichern den "Development Service" um weitere Funktionalitäten an und **erhöhen damit den Gesamtmehrwert**.
- Die konsumierenden Services lagern die Technologieverantwortung an den "Development Service" aus und **reduzieren damit Komplexität und Risiko**.

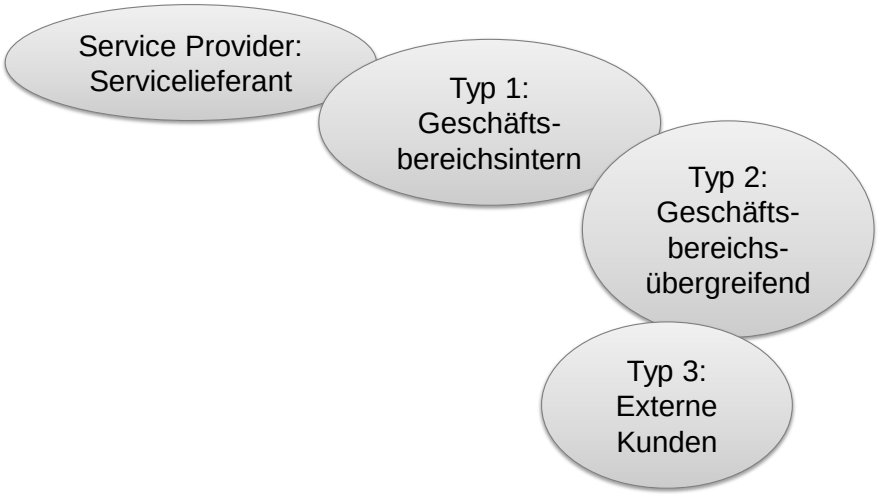


3.3 IT-Dienstleister (IT Service Provider)

Laut ITIL ist ein Dienstleister (engl.: Service Provider) eine Organisation, die einem oder mehreren Kunden Dienstleistungen zur Verfügung stellt. Es wird hierbei zwischen drei Typen von Dienstleistern unterschieden:

- **Typ 1**: Ein Interner Dienstleister (engl.: Internal Service Provider) ist Teil des Geschäftsbereiches für den die Dienstleistungen erbracht werden.
- **Typ 2**: Ein Interner gemeinsam genutzter Dienstleister (engl.: Internal Shared Service Provider) stellt gemeinsam genutzte Dienstleistungen für mehr als einen Geschäftsbereich bereit.
- **Typ 3**: Ein Externer Dienstleister (engl.: External Service Provider) stellt Dienstleistungen für externe Kunden zur Verfügung.

Definition „Service Provider“



Der "Kunde" möchte durch intelligenten Einsatz der IT eine höchstmögliche Qualitätsverbesserung mit gleichzeitiger Kostensenkung erzielen. Demzufolge besitzt und betreibt er eine, auf seine Geschäftsprozesse abgestimmte, IT-Lösung. Diese IT-Lösung / Anwendung wird durch eine Anwendungsbetreuung dauerhaft betreut und weiterentwickelt. Die Betreuung der Anwendung erfolgt entweder durch eine interne IT-/EDV-Abteilung (Typ 2) oder durch einen externen IT-Dienstleister (Typ 3). In beiden Fällen handelt es sich bei der Tätigkeit um eine IT-Dienstleistung bzw. bei der realisierenden Organisation und einen IT-Dienstleister.

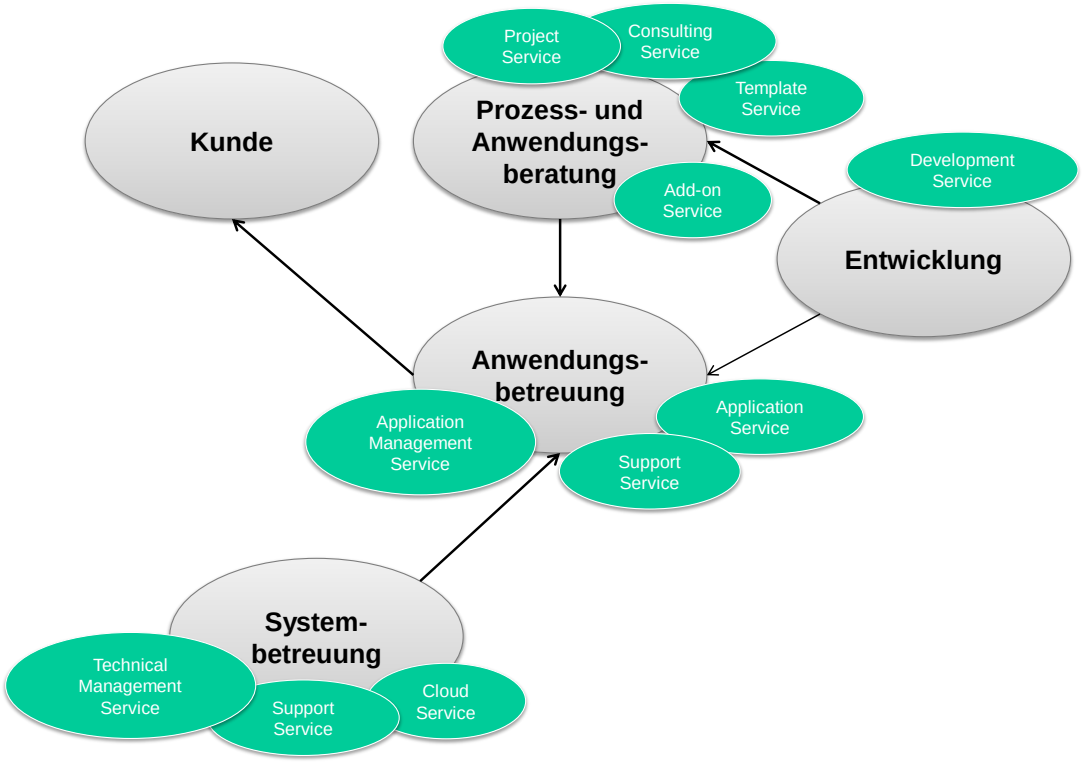
Die Anwendungsbetreuung ist in der Lage, den dauerhaften Betrieb der IT-Lösung sicherzustellen indem sämtliche Störungen aufgenommen, analysiert und behoben sowie kleinere Weiterentwicklungen realisiert werden. Die Einführung komplett neuer Funktionen oder der Umbau der IT-Lösung (inkl. Upgrade auf eine neue Version der Standardlösung) benötigt jedoch andere Fähigkeiten, z.B. Knowhow zu neuesten Technologien, und andere Rollen, z.B. Projektleiter. Diese Tätigkeiten, inkl. der damit verbundenen Kosten und Risiken für Aus- und Weiterbildung bzw. Ressourcenauslastung, werden daher an die entsprechend spezialisierte Prozess- und Anwendungsberatung übertragen.

Die Prozess- und Anwendungsberatung kann Teil derselben Abteilung wie die Anwendungsbetreuung sein (Typ 1). Es sollten jedoch, wegen der verschiedenen Skills + Verfügbarkeiten und zur Sicherstellung des 4-Augen-Prinzips bei der Übergabe in die Produktion, stets verschiedene Personen zum Einsatz kommen. Da die Prozess- und Anwendungsberatung im Gegensatz zur Anwendungsbetreuung nicht permanent benötigt wird, wird es sich hierbei jedoch meistens um einen externen IT-Dienstleister (Typ 3) handeln.

Die Prozess- und Anwendungsberatung analysiert und optimiert Geschäftsprozesse, sie sammelt die Kundenanforderungen ein und erstellt darauf basierend ein Konzept welches anschließend durch Konfiguration der IT-Lösung umgesetzt wird. Hierbei greift die Prozess- und Anwendungsberatung in besonders techniklastigen Projekten, in denen die Möglichkeiten der Konfiguration nicht ausreichen, auf die Anwendungsentwicklung zurück. Es kann sich dabei um technisch versierte Anwendungsberater handeln (Typ 1) oder um Ressourcen einer eigenen Anwendungsentwicklungsabteilung (Typ 2) oder sogar um einen externen IT-Dienstleister (Typ 3).

Die Anwendungsentwicklung ist damit in der Regel für die Anwendungsberatung tätig. Der Vollständigkeit halber sei erwähnt, dass die Anwendungsentwicklung für die Analyse und Beseitigung von Programmfehlern bzw. für das Tätigen kleiner Weiterentwicklungen auch direkt für die Anwendungsbetreuung aktiv werden kann. Wie auch immer: Die Anwendungsentwicklung ist damit stets ein Dienstleister (meistens vom Typ 2) und ausnahmslos für die IT-Branche tätig, da die Kunden (Anwendungsberatung und -betreuung) stets dieser Branche zuzuordnen sind. Bei der Dienstleistung "Entwicklungsberatung und Programmierung" handelt es sich zweifelsohne um eine IT-Dienstleistung, da hier Mittel der Informationstechnologie zum Einsatz kommen.

Der Vollständigkeit halber sei erwähnt, dass die Anwendungsberatung mehrere Dienstleistungen anbietet: Konkrete Add-ons, Konkrete Anwendungsberatungsleistung, Konkrete Projekte (Ein Projekt ist ein Service!). Auch sei erwähnt, dass es zahlreiche weitere hier nicht genannte IT-Dienstleister geben kann. So wird das IT-Hosting und die Systembetreuung meist an einen separaten IT-Dienstleister ausgelagert (engl.: Outsourcing). Die Prozess- und Anwendungsberatung kann u.U. in eine Prozessberatung und eine Anwendungsberatung "zerfallen". Und es gibt weitere IT-Dienstleister, die, wie die Anwendungsentwicklung, in der dritten Reihe stehen. Als Beispiel seien Beratungen für die Technologien SAP Netweaver Portal, Business Intelligent, Process Integration, Master Data Management, Identity Management, Solution Manager und viele mehr genannt. Auch nicht vergessen werden darf der Vertrieb mit seinen Sales Services, der ja sozusagen in der ersten Reihe steht und auf den wir aus Vereinfachungsgründen an dieser Stelle ebenfalls nicht genauer eingehen können. Und dann gibt es noch Dienstleistungen und Dienstleister für die zahlreichen Hard- und Softwareressourcen Anwendungsserver, Datenbank, Netzwerk, Festplatte, Strom, ..., die, da sie das Fundament für alle anderen Dienstleistungen bilden, üblicherweise als erstes mit IT Service Management in Verbindung gebracht werden und auf die wir hier aus Komplexitätsgründen ebenfalls nicht genauer eingehen konnten.



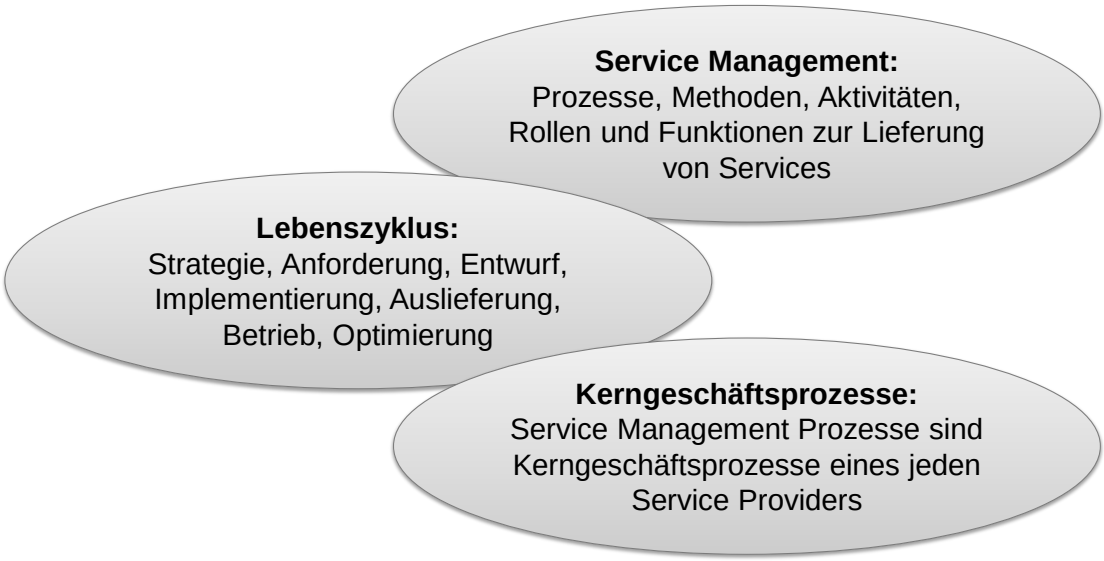
3.4 Steuerung und Verwaltung von IT-Dienstleistungen (IT Service Management)

Gemäß ITIL ist das Service Management die für Generierung und Betrieb von Dienstleistungen verfügbare und notwendige Gesamtheit der spezialisierten organisatorischen Fähigkeiten. Oder anders gesagt: Ein jeder (IT-)Dienstleister benötigt bestimmte Prozesse, Methoden, Aktivitäten, Rollen und Funktionen (Abteilungen) um Dienstleistungen anbieten und betreiben zu können. Die Gesamtheit dieser "Fähigkeiten" bezeichnen wir als (IT) Service Management.

Der durch das Service Management abgedeckte Dienstleistungslebenszyklus umfasst die Phasen "Strategie" (engl.: Strategy), "Anforderung" (engl.: Requirement), "Entwurf" (engl.: Design), "Implementierung" (engl.: Build), "Auslieferung" (engl.: Deployment), "Betrieb" (engl.: Operation) und "Optimierung" (engl.: Continual Improvement). Die Phasen "Anforderung", "Entwurf", "Implementierung" und "Auslieferung" werden üblicherweise bei ITIL unter dem Begriff Überführung (engl.: Transition) zusammengefasst.

An dieser Stelle seien folgende Aussagen besonders hervorgehoben: 1. Die (IT) Service Management Prozesse sind die Kerngeschäftsprozesse eines jeden (IT) Service Providers. 2. Es existieren weitere Haupt- und Nebenprozesse (z.B. Vertrieb, Partnermanagement, Personalabrechnung). 3. Das IT Service Management umfasst nicht nur Prozesse, sondern weitere "spezialisierte organisatorische Fähigkeiten" wie Methoden, Aktivitäten, Rollen und Funktionen.

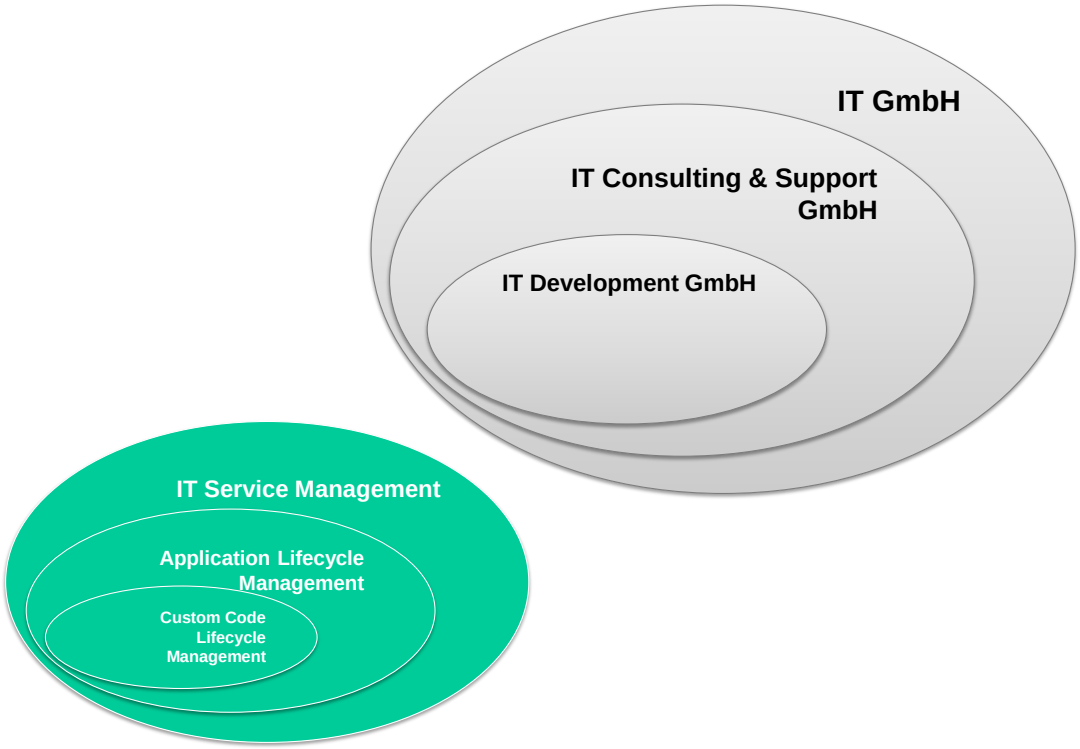
Definition „Service Management“



Das IT Service Management nach ITIL beschreibt unter anderem die Kerngeschäftsprozesse eines jeden IT-Dienstleisters. Angenommen, die Anwendungsentwicklungsabteilung wäre ein eigenständiges externes Unternehmen und damit ein eigenständiger externer IT-Dienstleister. Dann würde sie zweifelsohne ein vollumfängliches IT Service Management leben (müssen). Demnach lebt auch eine interne Anwendungsentwicklungsabteilung ein vollumfängliches IT Service Management, da hinsichtlich des IT Service Managements keine wirklichen Unterschiede zwischen internen und externen IT-Dienstleistern existieren. Da es sich in sämtlichen Fällen um einen speziellen IT-Dienstleister handeln würde, wäre auch das IT Service Management eine Spezialisierung / Konkretisierung des bekannten generischen ITIL-Frameworks.

In der Vergangenheit wurde häufig eine ausschließliche Fokussierung des IT Service Management auf den Betrieb unterstellt. Gleichzeitig wurde angenommen, dass für Anwendungsdienstleistungen als weiteres Framework das Application Lifecycle Management existiert, welches teilweise Überschneidungen zum IT Service Management besitzt, teilweise aber auch disjunkte Inhalte umfasst. Das ist spätestens seit ITIL V3 nicht mehr korrekt: Das Application Lifecycle Management ist "nur" eine Spezialisierung / Konkretisierung des IT Service Managements für eine bestimmte Gruppe von IT-Dienstleistern, nämlich Anwendungsberatern und -betreuern.

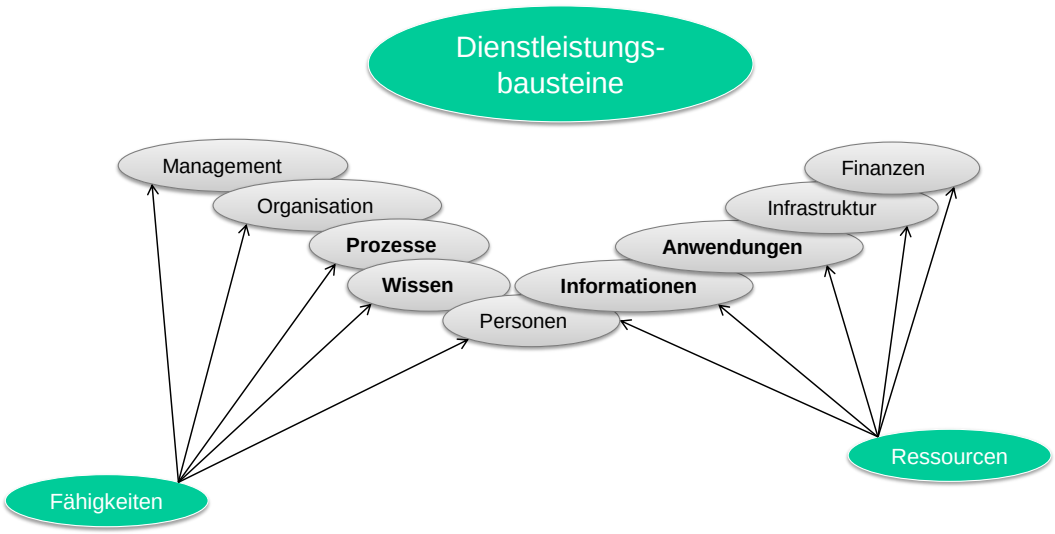
Die Anwendungsentwicklung wiederum ist eine Spezialisierung der Anwendungsberatung und -betreuung. Damit bedarf das IT Service Management einer erneuten Spezialisierung / Konkretisierung...zum **Custom Code Lifecycle Management**.


IT GmbH
IT Consulting & Support GmbH
IT Development GmbH
IT Service Management
Application Lifecycle Management
Custom Code Lifecycle Management

4 Bausteine von IT-Dienstleistungen (IT Service Assets)

Die für die Erbringung von Dienstleistungen benötigten Ressourcen und Fähigkeiten einer Organisation, bilden die Grundlagen, Ausgangsstoffe bzw. Bestandteile (engl.: Service Assets) dieser Dienstleistungen. Die Bestandteile "Finanzen", "Infrastruktur", "Anwendungen", "Informationen" und "Personen" sind "relativ leicht erwerbbare, direkt greifbare" Elemente, die zur Erbringung eines Services beitragen können und zählen damit zu den Ressourcen (engl.: Ressources). Dagegen gehören "Management", "Organisation", "Prozesse", "Wissen" und "Personen" zu den zur Durchführung einer Serviceaktivität benötigten Fähigkeiten (engl.: Capability) einer Organisation, einer Person, eines Prozesses, einer Anwendung, eines Configuration Items oder einer IT Dienstleistung und sind weder leicht erwerbbar, noch direkt greifbar.

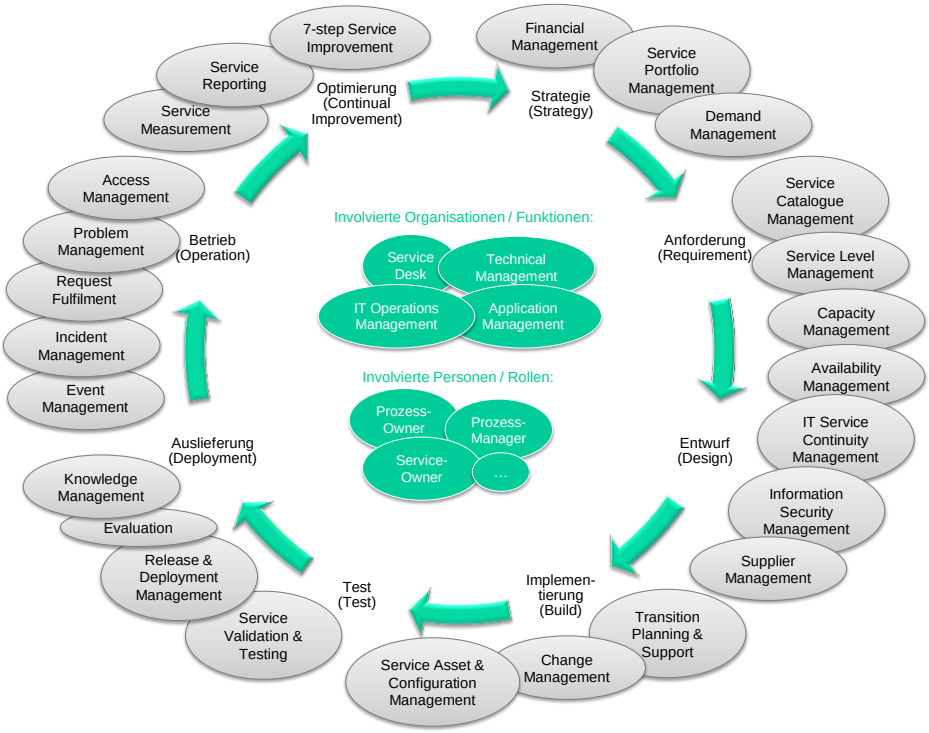
Definition „Service Asset“



4.1 Dienstleistungsbaustein "Prozesse" (IT Service Asset "Prozesses")

Laut ITIL V3 umfasst das IT Service Management fünfundzwanzig Prozesse, welche durch vier Organisationen / Abteilungen / Funktionen gelebt werden. Zu jedem Prozess existiert ein Prozess Owner und mehrere Prozess Manager.

Für detaillierte Informationen zu den Funktionen, Rollen und Prozessen im Rahmen des IT Service Managements sei auf einschlägige offizielle Informationsquellen, wie z.B. die Kernbücher der IT Infrastructure Library, verwiesen.



Nehmen wir einmal an, ein Anwendungsentwickler fällt krankheitsbedingt unerwartet aus und ein aktuell laufendes Projekt ist hierdurch gefährdet. Was tun Sie als Projektleiter? Richtig, Sie melden diese Störung an den Entwicklungsleiter und machen damit implizit eine Störungsmeldung auf.

Bei Störungen an der Hardware oder Software würden wir ohne zu zögern eine Störungsmeldung (engl.: Incident) eröffnen. Warum nicht auch bei Personen (engl.: Human Ressources)? Möglich, dass Ihr Unternehmen noch relativ überschaubar ist und Sie sich daher in Ihrer Rolle als Projektleiter stets direkt an den Entwicklungsleiter wenden können. Bei größeren und komplexeren Abteilungsstrukturen ist das nicht mehr möglich bzw. sinnvoll. Außerdem ist es auch egal, ob Sie einen Incident eröffnen, eine E-Mail schreiben oder zum Telefonhörer greifen. Sie initiieren damit definitiv einen Incident und leben den Incident Management Prozess.

Gleiches gilt für alle anderen Aktivitäten im Zusammenhang mit der IT-Dienstleistung "Entwicklungsberatung und Programmierung": Jede Aktivität lässt sich einem IT Service Management Prozess zuordnen [z.B. Incident Management (INC), Supplier Management (SUP), Problem Management (PROB), Change Management (CHA), Capacity Management (CAP), Service Validation & Testing (VT), Release & Deployment Management (RD), ...] und andersherum: Zu jedem IT Service Management Prozess lassen sich konkrete Aktivitäten finden.

Wenn Sie sich in Ihrem Unternehmen dieser wichtigen Tatsache bewusst sind, dann haben Sie den Übergang zu einem bewusst gelebten IT Service Management in Ihrer Anwendungsentwicklungsabteilung bereits geschafft. Inwieweit Sie die Prozesse und Aktivitäten formalisieren und reglementieren wollen und sollten, steht dabei auf einem anderen Blatt und ist sehr stark von Ihrer Unternehmensgröße und der Unternehmenskultur abhängig.

INC (O)
- Ein Entwickler fällt krankheitsbedingt unerwartet aus.
- Ein aktuell laufendes Projekt ist gefährdet.

SUP (O)
- Als Workaround wird ein externer Entwickler eingekauft.
- Das Projekt kann termingerecht fortgesetzt werden.

PROB (O)
- Der Entwickler fällt voraussichtlich längere Zeit aus.
- Mehrere geplante Projekte sind damit gefährdet.

CHA (CI)
- Ein anderer Entwickler durchläuft eine Weiterbildungsmaßnahme.
- Er übernimmt zukünftig die Aufgaben des ausgefallenen Entwicklers.

CAP (D)
- Aufgrund eines permanenten Ressourcenengpasses wird eine weitere Entwicklerplanstelle geschaffen und ausgeschrieben.

CHA (B)
- Es werden mehrere Bewerbungsgespräche geführt und ein neuer Entwickler eingestellt.
- Dieser neue Entwickler durchläuft ein umfangreiches Ausbildungsprogramm.

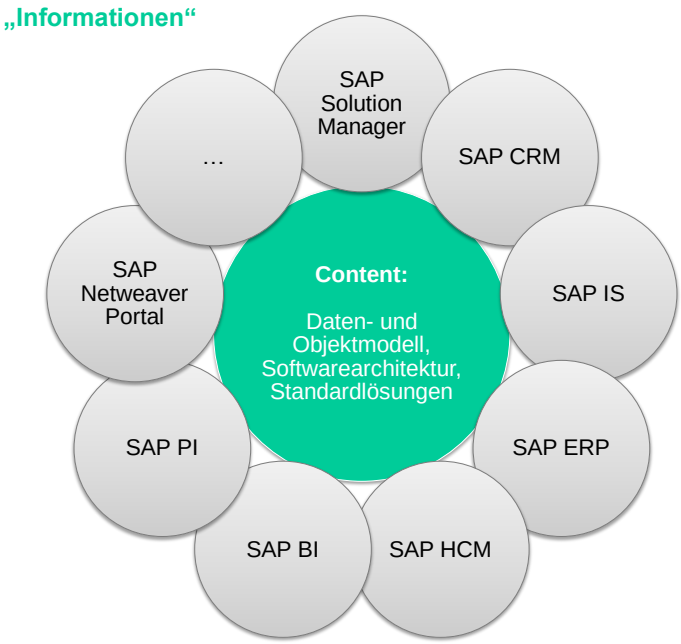
VT RD
- Nach seiner Einarbeitung / Bewährung steht der neue Entwickler mit Ablauf der Probezeit als reguläre Entwicklerressource zur Verfügung.

4.2 Dienstleistungsbaustein "Informationen" (IT Service Asset "Information")

Informationen zählen als Ressourcen zu den wichtigsten Bestandteilen von Dienstleistungen. Sie sind relativ leicht in Form von Onlinedokumentationen, Büchern, Schulungsunterlagen, Forumsbeiträgen etc. erwerbbar und nutzbar. ITIL selbst macht keinerlei Vorgaben zu diesem Dienstleistungsbestandteil, da er abhängig von der konkreten Dienstleistung ist und sehr verschieden ausfallen kann.

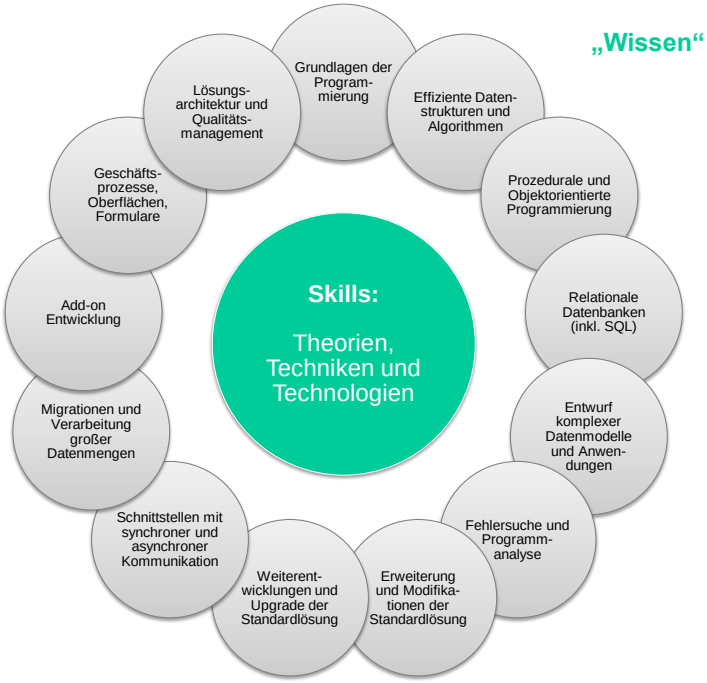
Für die Anwendungsentwicklung im Umfeld der SAP Business Suite ist mit "Informationen" der Inhalt (engl.: Content) der SAP Systeme gemeint. Das beginnt mit dem konkreten Daten- und Objektmodell (Tabellen und deren Beziehungen, Klassen / Objekte und deren Beziehungen, Funktionsbausteine und Methoden, …), geht über die aus mehreren Schichten bestehende Softwarearchitektur (Funktionsbausteine ohne Pufferung, mit Pufferung, Webservices, User Interfaces, Workflows, Business Add-ins, Erweiterungsspots, …) bis hin zu kompletten Standardlösungen (Transaktionen, Reports, …).

Die Fülle an Informationen ist unter technischen Gesichtspunkten in zahlreiche Systeme, Module bzw. Anwendungsgebiete untergliedert und auf zahlreiche Informationsmedien verteilt. Als Anwendungsentwickler ist es nützlich diese Informationen auswendig zu können. Es reicht jedoch in der Praxis vollkommen aus, erstens nur Informationen zu den Anwendungsgebieten abrufbereit zu haben, in denen man auch tatsächlich unterwegs ist, und zweitens zu wissen, wo es steht.



4.3 Dienstleistungsbaustein "Wissen" (IT Service Asset "Knowhow")

Wissen zählt zu den Fähigkeiten und damit zu den nicht ganz so leicht erwerbbaren Dienstleistungsbestandteilen. Der Grund hierfür ist die Tatsache, dass Wissen zwar auf Informationen basiert, jedoch zu einem noch größeren Teil durch Erfahrung und Talent beeinflusst wird: Ein Anwendungsentwickler kann eine ABAP Schulung inkl. Zertifizierung erfolgreich durchlaufen und wird dennoch niemals ein guter Anwendungsentwickler sein, wenn er keine theoretischen Kenntnisse zu effizienten Datenstrukturen und Algorithmen, keine Praxiserfahrung in der Programmierung inkl. Nutzung unterstützender Werkzeuge und kein logisches, mathematisches, algorithmisches Denken besitzt.



Die folgende Liste enthält ohne Anspruch auf Vollständigkeit das "komplette" Wissen der ABAP Anwendungsentwicklung. Das Wort "Wissen" steht hier synonym für Fähigkeiten (engl.: Skills). Gemeint sind damit Kenntnisse und Erfahrungen im Umgang mit den gängigen Theorie der Programmierung, verschiedenen Programmiertechniken und sämtlichen genutzten SAP Netweaver Technologien und Werkzeugen.

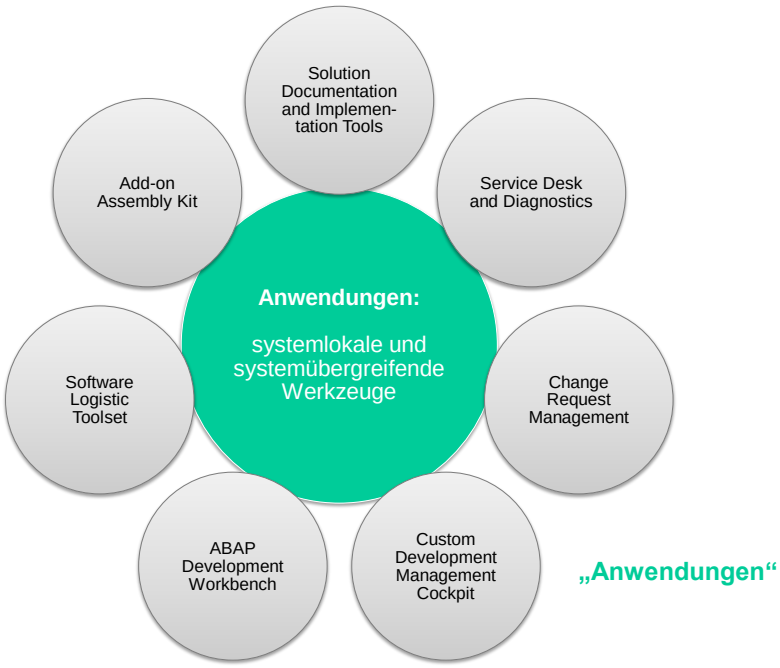
Skill	Beschreibung
Grundlagen der Programmierung	Welche Konzepte der Programmierung gibt es? Was sind Gemeinsamkeiten, Unterschiede und Einsatzgebiete von funktionaler, logischer, prozeduraler und objektorientierter Programmierung? Was sind Bedingungsblöcke, Schleifen, Unterprogramme, ...? Was bedeutet iterativ, rekursiv, sequentiell, parallel, ...? Was ist der Unterschied zwischen relationalen und objektorientierten Datenbanken? Was ist das Model-View-Controller-Entwurfsmuster (MVC)? Was ist eine Client-Server-Architektur und was bedeutet SOA (Service-orientierte Architektur) für meine Anwendung? Welche Modelle der Softwareentwicklung [Wasserfallmodell, V-Modell, Scrum...] gibt es?
Effiziente Datenstrukturen und Algorithmen	Welche Datenstrukturen und Algorithmen gibt es [Stacks, Queues, Heaps, Felder, Vorwärts und rückwärts verkettete Listen, Sortier- und Suchalgorithmen]? Welche sind für welches Einsatzszenario geeignet? Wie können Algorithmen entwickelt, analysiert und bewertet werden? Wie funktionieren Laufzeit- und Speicheranalyse?
Prozedurale und Objektorientierte Programmierung	Wie funktioniert die Programmiersprache ABAP? Wie sind die Konzepte der prozeduralen Programmierung mit ABAP/4 [ABAP Development Workbench, Pakete, Funktionsgruppen, Funktionsbausteine, Unterprogramme, Transaktionen, Reports, Businessobjekte, SAP Easy Access Menüs, ...]? Wie sind die Konzepte der objektorientierten Programmierung mit ABAP Objects [Klassen, Methoden, Interfaces, ...]?
Relationale Datenbanken (inkl. SQL)	Wie sind die allgemeingültigen bzw. sichtbaren Konzepte von durch SAP unterstützten relationalen Datenbanken [Indizierung, Sperrkonzept, Transaktionskonzept, ...] und wie erfolgt der Zugriff mit Open SQL [Select, Insert, Delete, Update, Subquery, Join, ...]? Wie funktioniert das Anlegen und Pflegen von Objekten des Data Dictionarys [Domains, Datenelemente, Tabellen, Texttabelle, Fremdschlüsselbeziehungen, Views, Viewcluster, Pflegedialoge, Suchhilfen, IMG-Leitfäden, ...]?
Entwurf komplexer Datenmodelle und Anwendungen	Wie erfolgen das Design komplexer Datenmodelle oder das Design und die Entwicklung komplexer Anwendungen? Wie funktionieren gängige Modellierungstechniken und -Werkzeuge [ER-Modell, UML, BPMN, ...]?

Fehlersuche und Programmanalyse	Welche Herangehensweisen zur Durchführen von Fehlersuchen und Programmanalysen haben sich bewährt? Wie funktionieren die verschiedenen Test-, Debugging-, Tracing-, Logging- und Monitoring-Werkzeuge [Unit Test Framework, ABAP-Debugger, SQL-Trace, Berechtigungstrace, ABAP-Trace, Dynpro-Trace, Smartform-Trace, RFC-Trace, Code-Scanner, ...]?
Erweiterung und Modifikation der Standardlösung	Wie erfolgen Veränderungen und Erweiterungen der SAP Standard Lösung mittels des SAP Enhancement & Modification Framework [Modifikationen, Customer Exits, Business Add-Ins, Explizite und Implizite Erweiterungen, Business Transaction Events, Event Handler & Callback Subscription Tables, Easy Enhancement Workbench (EEWB), Application Enhancement Tool (AET) Business Data Toolset (BDT), ...]?
Weiterentwicklungen und Upgrade der Standardlösung	Wie ist ein Upgrade der SAP Lösung und eine kontinuierliche Verbesserung der Kundenlösung (inkl. Rückführung auf den Standard) durchzuführen? [Solution Browsers, Business Function Prediction Services (BFP), Product Available Matrix (PAM), SAP Solution Manager Maintenance Optimizer, SAP Software Logistic Toolset, SAP Enhancement & Switch Framework, SAP Note Assistant, Modification and Enhancement Adjustment, SAP Solution Manager Clearing Analysis, SAP Solution Manager Upgrade/Change Impact Analysis, ...]
Schnittstellen mit synchroner und asynchroner Kommunikation	Wie entwickelt man Schnittstellen [BAPI, RFC, tRFC, qRFC, bgRFC, Web-Services] für einen synchronen oder asynchronen Datenaustausch? Wie erfolgen standardisiertes + kontinuierliches Einpacken, Versenden, Empfangen und Auspacken von Daten zur asynchronen Kommunikation innerhalb der SAP-Lösung oder zwischen der SAP Lösung und externen Systemen [BDOC, IDOC, ALE]? Welche Möglichkeiten zur Konfiguration und Anwendungsentwicklung im Umfeld der Datendrehscheibe "SAP Process Integration / SAP Exchange Infrastructure" gibt es?
Migrationen und Verarbeitung großer Datenmengen	Wie erfolgen Migrationen von Daten [Batch-Input, Legacy System Migration Workbench, Data Exchange Workbench, ...] und wie ist eine effiziente Verarbeitung großer Datenmengen [Einlesen CSV-/XML-Datei, Parallelisierung, Protokollierung, ...] möglich?

Add-on Entwicklung	Wie erfolgt die Anwendungsentwicklung von ausliefer- und zertifizierungsfähigen ABAP Add-ons? Wie funktioniert das Prüfen und Verpacken von Add-ons mit dem Add-on Assembly Kit (AAK)? Wie erfolgt die Installation von Add-ons mit dem SAP Software Logistic Toolset?
Geschäftsprozesse, Oberflächen, Formulare	Wie erfolgt die Anwendungsentwicklung von SAP Business Workflows und Business Process Management Prozessen. Wie erfolgt die Anwendungsentwicklung von SAP-GUI-Oberflächen [ABAP Reports oder ABAP Dynpro, mit komplexen Selektionsbildern, unter Nutzung von ALV-Grids und anderen EnjoySAP-Controls]? Wie erfolgt die Anwendungsentwicklung von Web-Oberflächen [Business Server Pages, SAP WebClient UI, WebDynpro für ABAP mit/ohne Floorplan Manager]? Wie erfolgt die Anwendungsentwicklung von (interaktiven) Formularen [SAP-Script, Smartforms, Interactive Adobe-Forms, Streamserve unter Verwendung der SAP IS-U Druckworkbench, Microsoft Word Formulare inkl. Makroprogrammierung und inkl. Erstellung / Ergänzung / Anbindung von Webservices zur Datenbeschaffung, Kommunikation mit Microsoft Office über die SAP Office Integration Schnittstelle und über die OLE-Schnittstelle]?
Lösungsarchitektur und Qualitätsmanagement	Gibt es Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation mit Vorgaben zu Entwicklungsstil, mit Angaben zu verbotenen und empfohlenen Programmbefehlen und mit Aussagen zu Softwarearchitektur und Qualitätssicherungsmaßnahmen? Gibt es Regeln und Hilfsmittel zur Quellcodeformatierung [Pretty Printer]? Gibt es Vorgaben zur Benamung und Dokumentation von Entwicklungsobjekten [Namensräume, Namenskonvention, Quellcodekopf, Programmdokumentation]? Wie erfolgt die Konfiguration und Nutzung von Prüfwerkzeugen [Syntaxprüfung, Code Inspector / Erweiterte Syntaxprüfung, Unit Test]? Wie funktioniert das SAP Änderungs- und Transportwesen [Transportarchitektur für Betrieb und Projekt, Transport Organizer, Transport Management System, SAP Solution Manager Change Request Management]? Wie ist mittels SAP Solution Manager eine vollständige, aktuelle und transparente Dokumentation der Kundenentwicklungen / Kundenlösung realisierbar [Solution Directory, Custom Code Data Base, ...]? Welchen konkreten Bedeutungen und Auswirkungen haben IT Service Management, Software Engineering und Projektmanagement auf Vorgehen und Ergebnis der Anwendungsentwicklung?

4.4 Dienstleistungsbaustein "Anwendungen" (IT Service Asset "Applications")

Wie bereits in den vorangegangenen Abschnitten erwähnt, stellt SAP zahlreiche Anwendungen und Werkzeuge für die Anwendungsentwicklung im Umfeld der SAP Business Suite zur Verfügung. Diese Werkzeuge automatisieren und unterstützen die im nächsten Kapitel ausführlich besprochenen Aktivitäten des Custom Code Lifecycle Managements und sind entweder in dem jeweiligen SAP System als Bestandteil der SAP Netweaver Basiskomponente lokalisiert (systemlokal) oder aber im SAP Solution Manager zentralisiert (systemübergreifend).



Ohne Anspruch auf Vollständigkeit enthält die folgende Tabelle eine Auflistung und Kurzvorstellung der wichtigsten Werkzeuge für die ABAP-Anwendungsentwicklung:

Name	Beschreibung

SAP Solution Manager Solution Directory / SAP Solution Manager Solution Database (SolDB) **SAP Solution Manager Landscape Database (LMDB)**	Die Dokumentation der Gesamtlösung im Sinne des IT Service Management Prozesses "Configuration Managements" bzw. ALM-Prozesses "Solution Documentation" erfolgt im Solution Directory des SAP Solution Managers. Diese Dokumentation umfasst u.a. Geschäftsprozessszenarien, Geschäftsprozesse und Prozessschritte, Schnittstellen, Konzepte + Dokumente, Testfälle, E-Learning-Materialien sowie zughörige Customizingeinstellungen und Entwicklungsobjekte. Die Landschaftsdaten werden in der SAP Solution Manager Landscape Database vorgehalten.
SAP Solution Manager Solution Documentation Assistant (SoDocA) **SAP Solution Manager Business Process Repository (BPR)** **SAP Solution Manager Business Process Change Analyzer (BPCA)** **SAP Solution Manager Business Blueprint (BBP)** **SAP Solution Manager Business Process Configuration (BPC)** **SAP Solution Manager eLearning Management (ELM)**	Der SAP Solution Manager Solution Documentation Assistant ist Bestandteil des ALM-Prozesses "Solution Implementation" und dort für Redokumentationszwecke (Reverse Business Process Documentation) vorgesehen, indem die genutzten Repositoryobjekte und vorhanden Customizing-, Stamm- und Bewegungsdaten analysiert werden und daraus mittels teilweise komplexer Regeln und auf Basis einer Referenzdokumentation (aus dem SAP Solution Manager Business Process Repository (BPR), von SAP Partnern oder der eigenen Lösungsdokumentation) abgeleitet wird, welche Geschäftsprozesse "gelebt" werden. Das Werkzeug kann auch zum Aufspüren "gelebter" undokumentierter Entwicklungen (= Dokumentationslücken) genutzt werden. Die Funktion SAP Solution Managers Business Process Change Analyzer als Bestandteil des ALM-Prozesses "Test Management" ermöglicht durch Analyse von Ausführungsstatistiken/-Traces die exakte Identifizierung der von einer Anwendungsentwicklung oder Customizingänderung betroffenen Prozesse / Testfälle und sollte immer dann zum Einsatz kommen, wenn aus Zeit-, Budget- und Ressourcen-gründen nicht alle Testfälle getestet werden können, dennoch eine höchstmögliche Testabdeckung und eine bestmögliche Risikominimierung sichergestellt werden soll. In Einführungs- oder Upgradeprojekten erfolgt die Konzeption im SAP Solution Manager Business Blueprint. Die Dokumentation im Sinne der Berater- und Entwicklerdokumentation erfolgt im Werkzeug "SAP Solution Manager Business Process Configuration". Im Rahmen des SAP Solution Manager eLearning Management ist die Anwenderdokumentation inkl. Erstellung und Ablage von E-Learning-Materialien möglich.

SAP Solution Manager Service Desk (SD) **Knowledge Articles (KA)** **SAP Note Search (NS)** **SAP Call Maintenance (CM)**	Die Fehlerbearbeitung inkl. Fehlermeldung vom bzw. Rückmeldung an den Meldenden im Rahmen des Integrationstests und das Melden + Bearbeiten von Störungen und Problemen im Sinne eines Incident und Problem Managements erfolgen im SAP Solution Manager Service Desk (und ist Bestandteil des ALM-Prozesses "Application Incident Management"). Es handelt sich hier um eine vollständige und flexibel konfigurierbare Workflow- und Callcenter-Lösung mit der Möglichkeit, eine unternehmenseigene Wissensdatenbank zu nutzen, SAP-Hinweise passend zum jeweiligen System zu suchen, SAP-Meldungen zu eröffnen + zu bearbeiten und mit anderen ServiceDesk-Tools zu kommunizieren.
Service Desk Interface (SDI) **SAP Solution Manager Root Cause Analysis (RCA)** **SAP Solution Manager Business Process Monitoring (BPMon)** **SAP Solution Manager End-User Experience Monitoring (EEM)**	Der SAP Solution Manager stellt im Rahmen des ALM-Prozesses "Root Cause Analysis (RCA)" die in den SAP Solution Manager Service Desk integrierte Funktion "Solution Manager Diagnostics" zur Verfügung, die für eine Fehlersuche in verteilten und mehrschichtigen Lösungslandschaften unverzichtbar ist. Diese Funktion verfolgt eine Geschäftsaktivität über mehrere Systeme, Schichten und Komponenten, sammelt hierbei sämtliche Tracing-, Logging- und Monitoring-Informationen ein und ist damit in der Lage, den Programmablauf End-To-End eindeutig nachzuvollziehen und hierbei Fehlverhalten oder Performanceprobleme schnell und genau zu identifizieren und zu lokalisieren. Durch Einsatz der Werkzeuge SAP Solution Manager Business Process Monitoring (Überwachung von Prozessschritten und Schnittstellen) und SAP Solution Manager End-User Experience Monitoring (Testweises Ausführen von Prozessen / Aktivitäten) ist eine proaktive Störungserkennung möglich, was für den Endanwender eine signifikante Verbesserung der gefühlten Lösungsqualität zur Folge hat und was der Anwendungsbetreuung die schnelle Behebung der Störung ermöglicht, bevor teure Folgeschäden und -kosten entstehen bzw. bevor der Endanwender die Störung überhaupt bemerkt.

SAP Solution Manager Change Request Management (ChaRM) **Critical Object Approval Process (COAP)** **Project Intersection Check (PIC)** **Downgrade Protection (DGP)** **Cross System Object Lock (CSOL)** **Retrofit (RF)** **Quality Gate Management (QGM)** **Enhanced Change and Transport System (CTS+)** **Configuration Validation (CV)**	Die IT Service Management Prozesse "Change Management" und "Release Management" bzw. der ALM-Prozess "Change Control Management" werden mittels SAP Solution Manager Change Request Management unterstützt und automatisiert. Es handelt sich hierbei um eine vollständige und flexibel konfigurierbare Workflow-Lösung, mit direkter Integration des SAP Änderungs- und Transportwesens. Neben den Workflow-Funktionalitäten umfasst das ChaRM u.a. die sehr nützlichen Funktionen Critical Object Approval Process (COAP) zur Genehmigung von kritischen Objekten inkl. Modifikationen, Project Intersection Check (PIC) und Downgrade Protection (DGP) zur Identifizierung, Pflege und Überwachung von Transportabhängigkeiten, Cross System Object Lock (CSOL) zur Verhinderung von Parallelanpassungen (und damit einhergehender Überholerprobleme) mittels systemüber-greifender Objektsperren für Anwendungsentwicklung und Customizing, Retrofit (RF) zum nahezu vollautomatischen Parallelabgleich von Anpassungen der Wartungs- und der Projektlandschaft, Quality Gate Management (QGM) zur zentralen Steuerung und Überwachung der Transport-aktivitäten, Enhanced Change and Transport System (CTS+) zur Transportanbindung von u.a. Java-Entwicklungen und Portal-Konfigurationen, Configuration Validation (CV) zur Über-wachung bzw. zum Rekonstruieren bzw. zum Vergleich / Validieren von Änderungsaktivitäten durch Customizing, Anwendungsentwicklung, Anpassungen von System/-Profilparametern, ...

SAP Solution Manager Custom Code Lifecycle Management Database (CCLM / CCDB)	Das SAP Solution Manager Custom Code Lifecycle Management (CCLM) dient als zentrale systemübergreifende Custom Code Lifecycle Management Database und umfasst ein Verzeichnis sämtlicher kundeneigenen Entwicklungen inkl. Versionierung, inkl. Zuordnung von Verantwortlichkeiten und inkl. Attributisierungen.
SAP Solution Manager Custom Development Management Cockpit (CDMC) **Upgrade/Change Impact Analysis (UCIA)** **Clearing/Using Analysis (CA)** **Change and Transport System Analysis (CTSA)**	Das SAP Solution Manager Custom Development Management Cockpit zentralisiert zahlreiche weitere Funktionen des ALM-Prozesses "Custom Code Management (CCM)" im SAP Solution Manager. Unter anderem umfasst es die Funktionen Upgrade / Change Impact Analysis (UCIA) zur Identifizierung von nach einem Upgrade der SAP-Lösung anzupassenden Eigenentwicklungen durch Analyse der Verwendung von geänderten SAP-Repositoryobjekten, Clearing / Using Analysis (CA) zur Ermittlung nicht mehr verwendeter und damit zurückzubauender Eigenentwicklungen u.a. durch Auswertung der Aufrufstatistiken/-Traces und Change and Transport System Analysis (CTSA) zur Analyse der Objekte eines Transportauftrages hinsichtlich der Nutzung und der Version in den Systemen.
SAP Solution Manager Custom Code Analysis (CCA) **Clone Finder (CF)** **Interface and Dynamic Call Analysis (IA)** **Modification Overview (MO)** **Custom Code Remote Compare (CCRC)**	Zur SAP Solution Manager Custom Code Analysis zählen folgende Funktionen: Clone Finder (CF) zur Identifizierung von ähnlichen Programmen welche somit das Potential zu einer Zusammenführung haben oder beim Upgrade synchron anzupassen sind, Interface and Dynamic Call Analysis (IA) zur Ermittlung dynamischer Funktionsaufrufe durch Analyse von Registrierungstabellen, BADIs, Interfaceverwendungen etc., Modification Overview (MO) als zentrale + systemübergreifende + mächtigere Alternative zum systemlokalen SAP Modification Browser zur Anzeige von Modifikationen und Erweiterungen der SAP-Standard-Lösung sowie Custom Code Remote Compare (CCRC) zum systemübergreifenden Vergleich bzw. zur Versionsnachverfolgung von Eigenentwicklungen.

ABAP Development Workbench (DWB) **ABAP Debugger (DBG)** **SAP Tracing, Logging, Monitoring and Reporting Toolset** **SAP Pretty Printer (PP)** **SAP Syntax Check (SC)** **SAP Extended Syntax Check (ESC)** **SAP Code Inspector (CI)** **SAP Modification & Enhancement Toolset (MET)**	Die ABAP Entwicklungsumgebung umfasst viele Einzeltransaktionen zum Anlegen und Bearbeiten von Entwicklungsobjekten, welche zum (großen) Teil in der für den Anwendungsentwickler wichtigsten Transaktion SE80 zusammengefasst sind. Darüber hinaus existieren zahlreiche Transaktionen / Funktionen die nicht der Anwendungsentwicklung, sondern dem Debugging (ABAP Debugger), Tracing (SQL-Trace, Berechtigungstrace, ABAP-/Run-Time-Trace, ...), Logging (Applicationlog, Systemlog, Shortdump-Analysis, ...), Monitoring (Process Overview, User Overview, Gateway Monitor, ...) und Reporting zuzuordnen sind, und die dem Anwendungsentwickler für die Analyse von kundeneigenen oder SAP-eigenen Entwicklungen unverzichtbar sind. Zahlreiche dieser systemlokalen Hilfswerkzeuge zum Tracing, Logging, Monitoring und Reporting werden von der Funktion SAP Solution Manager Root Cause Analysis verwendet. Weitere Funktionen: SAP Pretty Printer dient der automatischen Formatierung des ABAP Quellcodes. SAP Syntax Check und SAP Extended Syntax Check (Transaktion SLIN) dienen der Prüfung des ABAP Quellcodes auf syntaktische Korrektheit. Der SAP Code Inspector (Transaktion SCI) dient der Entwicklungsanalyse, die deutlich über die normale bzw. erweiterte Syntaxprüfung hinausgeht. So ist der SAP Code Inspector u.a. in der Lage, die Anwendungsentwicklung speziell unter den Aspekten Performance, Mehrsprachenfähigkeit, Namenskonventionen, Programmierstil (veraltete Befehle), semantische Korrektheit (erreichbarer Code), etc. zu analysieren und damit potentielle Fehler oder Qualitätsmängel frühzeitig aufzudecken.
SAP Note Assistant (SNOTE) **SAP Modification Adjustment (SPDD + SPAU)** **SAP Enhancement Adjustment (SPAU_ENH + WCF_CC)** **SAP Modification Browser (MB)** **SAP Unicode Checker (UC)**	Der SAP Note Assistant (Transaktion SNOTE) dient dem Einspielen von SAP-Hinweisen zur Vorabkorrektur von Fehlern an SAP-Standard-Funktionalitäten. Die Werkzeuge zum Modifikations- und Erweiterungsabgleich (Transaktionen SPDD für Dictionaryobjekte, SPAU für Code-Modifikationen, SPAU_ENH für Erweiterungen und WCF_CC für Erweiterungen im SAP Web Client UI) dienen dem Abgleich von Veränderungen oder Erweiterungen des SAP-Standards im Rahmen eines Upgrades (Support Package Einspielung, Enhancement Package Einspielung, Releasewechsel). Angezeigt werden können die Modifikationen (inkl. Hinweise) und Erweiterungen (inkl. Customer-Exits, BADIs) im SAP Modification Browser (Transaktion SE95). Der SAP Unicode Checker (Transaktion UCCHECK) dient der Überprüfung der Unicodefähigkeit sämtlicher Eigenentwicklungen. Da hierzu eine Syntaxprüfung über alle Eigenentwicklungen durchgeführt wird, kann dieses Werkzeug auch unabhängig von einer

Enhancement & Switch Framework (SF) **SAP Test Environment (TE)** **SAP Test Data Directory (TDD)**	Unicodeumstellung zur Identifizierung von aufgrund von Syntaxfehlern definitiv anzupassenden Eigenentwicklungen nach einem Upgrade genutzt werden. Funktionale Erweiterungen werden in Form von Business Functions + Technical Usages über Enhancement Packages ausgeliefert und mittels Enhancement & Switch Framework (Transaktion SFW5) aktiviert. Diese Technik kann auch für kundeneigene Entwicklungen genutzt werden, um Einspielung und Aktivierung voneinander zu entkoppeln.
SAP Unit Test Framework (UTF) **SAP extended Computer Aided Test Tool (eCATT)**	Für den Komponententest stellt die ABAP Entwicklungsumgebung eine leistungsfähige Testumgebung inkl. Testdatenverzeichnis und der Möglichkeit zur Erstellung von automatischen Unittests zur Verfügung. Darüber hinaus wird die Erstellung und Ausführung automatisierter Tests durch das SAP extended Computer Aided Test Tool und das Component Based Test Automation Framework unterstützt.
Component Based Test Automation Framework (CBTA) **Customizing Comparison and Distribution (CCD / CSCOUT)**	Die Funktion Customizing Comparison and Distribution (CCD / CSCOUT) dient dem systemübergreifenden manuellen und automatischen Vergleich und Abgleich von Customizing-einstellungen und kann für Template-Rollout-Szenarien oder im Zusammenhang mit Stammdatenreplikationen genutzt werden.

SAP Not Assistant (SNOTE) **SAP Software Logistic Toolset (SLT)** **SAP Support Package Manager (SPAM)** **SAP Add-on Installation Tool (SAINT)** **SAP Enhancement Package Installer (SAPehpi)** **SAP Upgrade Tool (SAPup)** **SAP Software Update Manager (SUM)** **SAP Solution Manager Maintenance Optimizer (MOPZ)** **SAP Upgrade Dependency Analyzer (UDA)**	Das Einspielen von Vorabkorrekturen erfolgt mittels SAP Note Assistant. Für das Einspielen von Support Packages (Minor Releases) dient der SAP Support Package Manager (Transaktion SPAM). Zum Einspielen von neuen Softwarekomponenten dient das SAP Add-on Installation Tool (Transaktion SAINT). Und zum Upgrade von Softwarekomponenten in Form von Enhancement Packages oder Releasewechsel (Major Release) ist der SAP Enhancement Package Installer (SAPehpi) bzw. das SAP Upgrade Tool (Transaktion SAPup) zu nutzen. Alternativ zu diesen Einzelwerkzeugen kann / sollte der SAP Software Update Manager (SUM) verwendet werden. Das Identifizieren und Herunterladen der benötigten Installationsdateien erfolgt mittels SAP Solution Manager Maintenance Optimizer (MOPZ). Während der Maintenance Optimizer die Kompatibilität der Softwarekomponenten innerhalb eines SAP-Systems gewährleistet, kann der Upgrade Dependency Analyzer genutzt werden, um Abhängigkeiten zwischen verschiedenen Systemen zu identifizieren.

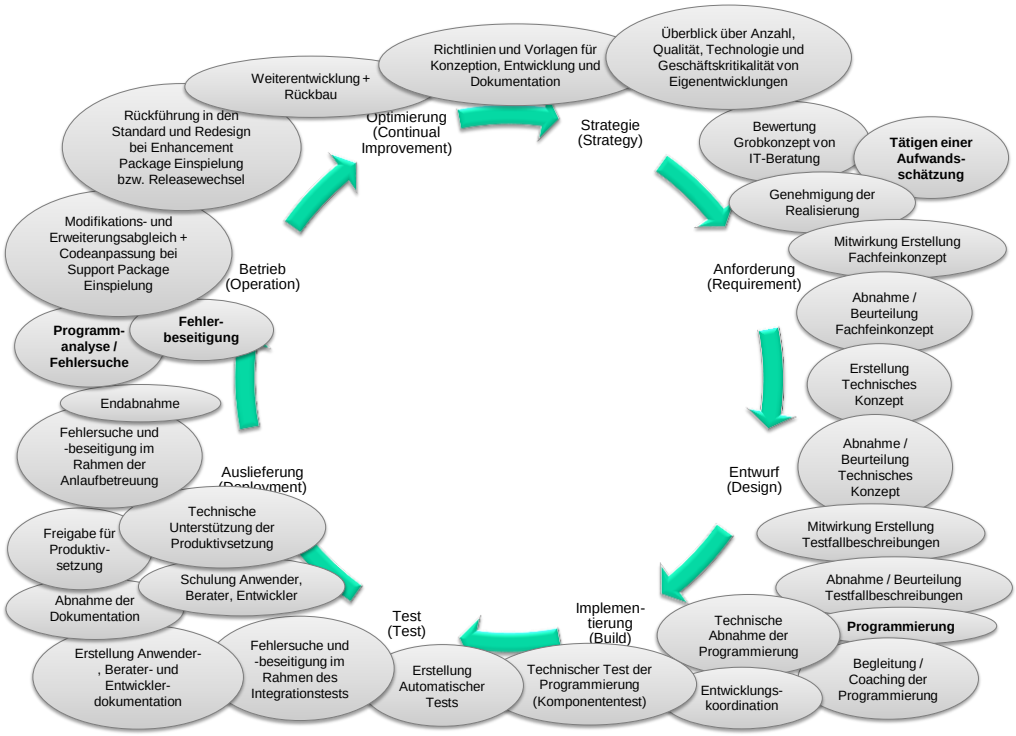
ABAP Add-on Assembly Kit (AAK) **SAP Software Delivery Composer (SSDC)** **SAP Software Delivery Assembler (SSDA)**	Das AAK ist ein Toolset zur Erzeugung auslieferungsfähiger Add-ons. Das Einspielen dieser Add-ons erfolgt mit dem SAP Software Logistic Toolset. Das AAK umfasst eine ausführliche Dokumentation mit zahlreichen Best-Practices zur Produkt / Add-on-Entwicklung und besteht im Wesentlichen aus zwei Funktionen: Der Software Delivery Composer (Transaktionscode SSDC) sammelt alle auslieferungs-relevanten Bestandteile einer Anwendungsentwicklung, prüft sie auf Konsistenz und Korrektheit und stellt sie zu einer Auslieferung zusammen. Der Software Delivery Assembler (Transaktionscode SSDA) packt die Auslieferung in ein einspielbares Paketformat. Außerdem können mit dem Software Delivery Assembler Einspielbedingungen definiert werden, welche die Voraussetzungen für ein korrektes Einspielen der Pakete schaffen.

Diese Übersicht stellt eine Zusammenstellung und Kurzvorstellung der wichtigsten für die Anwendungsentwicklung relevanten systemlokalen und systemübergreifenden Werkzeuge dar. Diese Übersicht ist keinesfalls vollständig. Eine Ergänzung und Detaillierung dieser Übersicht ist für ein umfassendes Verständnis bzw. zur Nutzung der Werkzeuge unerlässlich, würde jedoch den Rahmen dieser Publikation sprengen, zumal eine einzelne Funktion (wie z.B. der SAP Code Inspector) bereits ein ganzes Buch füllen kann. Zudem kommen "täglich" neue Funktionen hinzu (z.B. systemübergreifender Verwendungsnachweis und systemübergreifende Syntax-/ Schnittstellenprüfung für RFC-Aufrufe) oder werden zusammengeführt bzw. im SAP Solution Manager zentralisiert (z.B. SAP Software Logistic Toolset, SAP Custom Development Management Cockpit oder SAP Testworkbench).

5 Zweck von IT-Dienstleistungen (IT Service Utility)

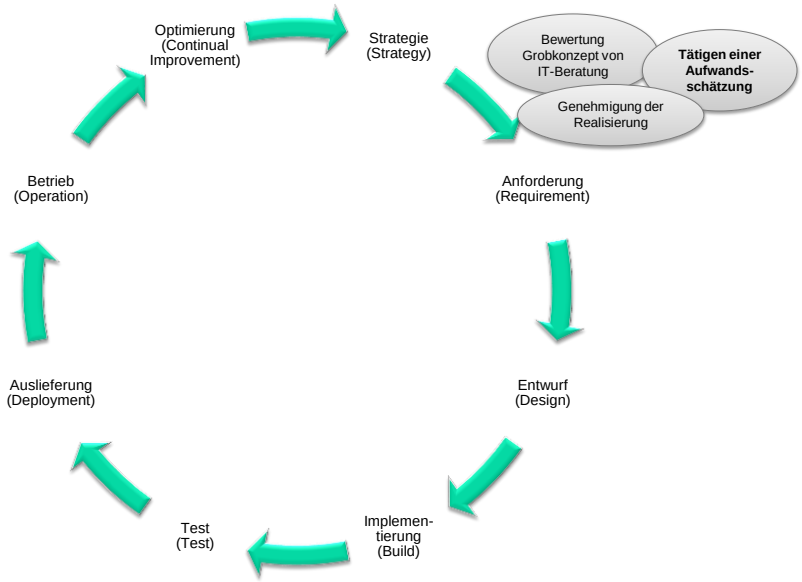
Jede Dienstleistung durchläuft die Phasen "Strategie", "Anforderung", "Entwurf", "Implementierung", "Test", "Auslieferung", "Betrieb" und "Optimierung". Da sich Dienstleistungen ständig verändern, werden diese Phasen nicht nur einmal sondern permanent / kontinuierlich durchlaufen, wobei hierbei in den einzelnen Phasen unterschiedlich lange verweilt wird und einige Phasen auch parallel durchlebt oder auch übersprungen werden können.

Der wirkliche Mehrwert für den Kunden entsteht durch die "Wertlieferung", welche ausschließlich in der Betriebsphase erfolgt und mit konkreten Aktivitäten und konkreten Ergebnissen verbunden ist. Das gilt für alle Dienstleistungen und somit auch für die Dienstleistung "Entwicklungsberatung und Programmierung". Der Wert der Dienstleistung "Entwicklungsberatung und Programmierung" besteht in der aktive Unterstützung und der damit einhergehenden signifikanten Wertsteigerung sämtlicher Phasen der Dienstleistungen der Anwendungsberatung und der Anwendungsbetreuung.



Im Folgenden werden die üblichen Aktivitäten der Anwendungsentwicklung gruppiert nach den unterstützten Dienstleistungsphasen vorgestellt. Diese Aktivitäten werden im SAP Netweaver Umfeld durch zahlreiche Tools der jeweiligen SAP-Systeme und des SAP Solution Managers unterstützt. Diese, im vorangehenden Kapitel bereits vorgestellten, Tools werden ohne Anspruch auf Vollständigkeit ebenfalls genannt.

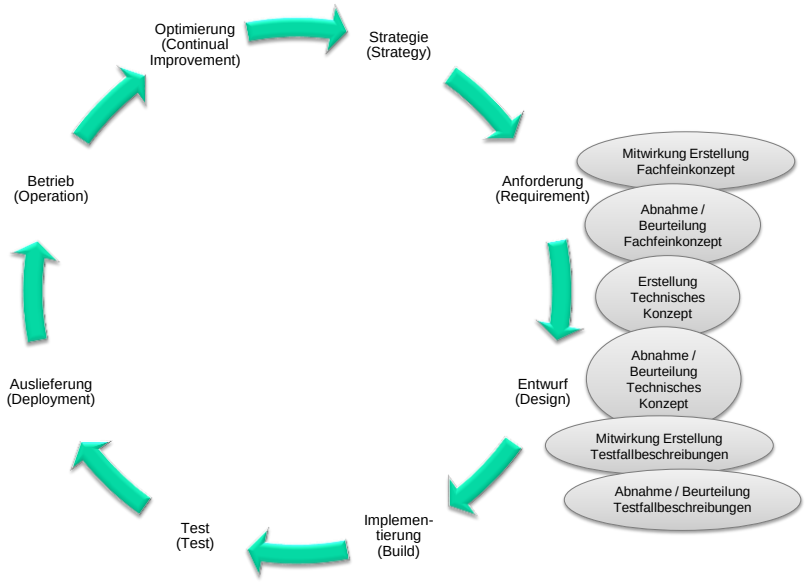
5.1 Anforderung (Requirement)



Phase	Aktivität	Beschreibung
Anforderung	**Bewertung Grobkonzept von IT-Beratung**	Meist ist bereits bei Aufnahme und Fixierung der Anforderung erkennbar, ob eine Realisierung im Standard möglich ist oder ob Entwicklungsbedarf besteht. Darüber hinaus bestehen am Anfang noch die meisten Freiheitsgrade, Einfluss auf die Lösung zu nehmen und eine standardnahe Realisierung zu ermöglichen. Das Grobkonzept (Lastenheft) wird üblicherweise von der Anwendungsberatung in Abstimmung mit dem Anwender/Kunden erstellt. Es ist jedoch sinnvoll bereits in dieser Phase die Anwendungs-entwicklung beratend einzubeziehen. Diese Aktivität sollte als Bestandteil der Change Management Phase "Bewertung eines Änderungsantrages" etabliert sein. Werkzeug: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).
	Tätigen einer Aufwand-schätzung	Mit der Grobkonzeption ist stets auch eine Abschätzung des Aufwandes für Konzeption, Umsetzung, Test, Dokumentation, Produktiv-setzung, Betrieb und Fehlerbeseitigung im Rahmen der Gewährleistung (komplett oder in Etappen) verbunden. Die Aufwandsschätzungen für die Entwicklungsbestandteile sind durch einen erfahrenden Anwendungsentwickler zu tätigen. Diese Aktivität ist immer durchführen. Sie hat großen Einfluss auf Umfang, Qualität und Realisierungszeitraum der Anwendungs-entwicklung. Zudem ist die Anwendungs-entwicklung an diese Schätzung gegenüber der Anwendungsberatung gebunden - Die Anwendungsentwicklung geht hierdurch einen Werkvertrag mit der Anwendungsberatung ein und muss sowohl für Überschreitung als auch für konkrete Korrekturen / Nachbesserungen einstehen. Werkzeug: SAP Solution Manager Change Request Management (ChaRM).

	Genehmigung der Realisierung **(QG 1)**	Bevor die Änderung realisiert wird und hierfür Budget und Ressourcen gebunden werden, ist eine explizite Genehmigung durch den Change Manager und weitere Entscheidungsträger im Rahmen der Change Management Phase "Genehmigung eines Änderungsantrages" erforderlich. An dieser Entscheidung ist auch die Anwendungsentwicklung beteiligt. Werkzeug: SAP Solution Manager Change Request Management (ChaRM).

5.2 Entwurf (Design)



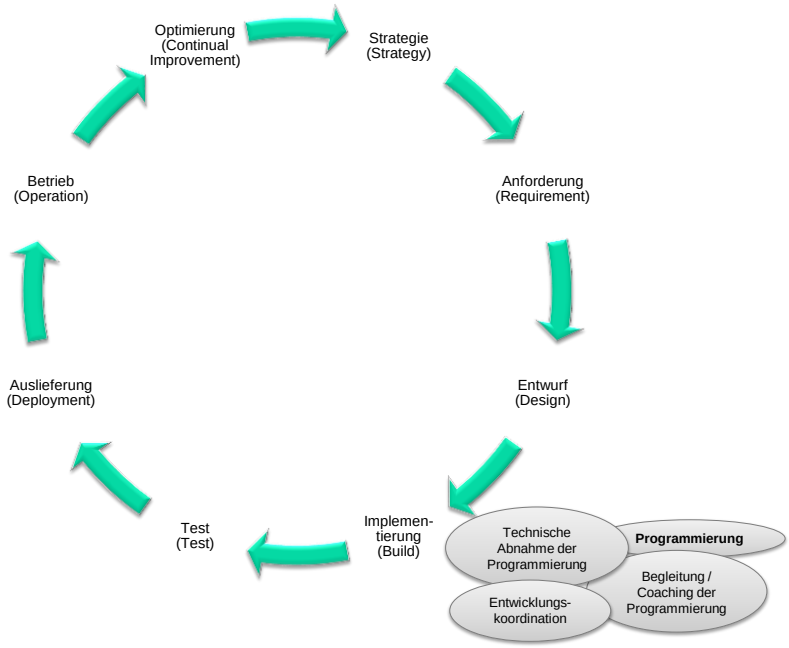
Phase	Aktivität	Beschreibung
Entwurf	**Mitwirkung Erstellung Fachfeinkonzept**	Da jede Lösungsimplementierung einen fachlichen Bezug besitzt und konfigurierbare Elemente umfasst, ist für die Erstellung des Fachfeinkonzeptes (Pflichtenheftes) in den meisten Fällen die Anwendungsberatung verantwortlich und zuständig. Die Mitarbeit der Anwendungsentwicklung ist dennoch zwingend notwendig: Zum einen ist die Anwendungs-entwicklung hier beratend tätig um zu vermeiden, dass dem Anwender/Kunden kostenintensiv bzw. fehleranfällig zu realisierende Funktion-alitäten versprochen werden. Zum anderen ist die Anwendungsentwicklung aktiv in die Erstellung einzubinden, weil einige Teile des Konzeptes bereits sehr technisch werden können und von der Anwendungsentwicklung zuzuarbeiten sind. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).
	Abnahme / Beurteilung Fachfeinkonzept **(QG 2)**	Zur Sicherung der Qualität der zu tätigenden Anwendungsentwicklung, ist im Sinne eines "Quality Gates" unbedingt eine Abnahme des Fachfeinkonzeptes durch einen erfahrenen Anwendungsentwickler durchzuführen. Dies ist umso notwendiger je weniger die Anwendungs-entwicklung in die Erstellung selbst einbezogen wurde, weil die Anwendungsberatung selbst gewisses technisches Knowhow besitzt, weil die Mitwirkung durch einen externen Anwendungs-entwickler realisiert wurde oder weil ein eher unerfahrener Anwendungsentwickler (Junior) an der Erstellung beteiligt war. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).

	Erstellung Technisches Konzept	Unabhängig davon, wie intensiv die Anwendungsentwicklung in die Erstellung des Fachfeinkonzeptes eingebunden wurde, gilt: Die Erstellung des Technisches Konzeptes (Spezifikation) ist alleinige Aufgabe der Anwendungsentwicklung...mit der kleinen Einschränkung, dass dies nur für die Entwicklungselemente gilt. Die Konfigurations-elemente (Konfigurationen der Standardlösung und der Entwicklungen) sind durch die Anwendungsberatung zu konzeptionieren. Die Anwendungsberatung sollte stets an der Technischen Konzeption mitwirken, um Verständnisfragen des Anwendungsentwicklers zügig beantworten bzw. Ideen des Entwicklers bewerten und ggf. nachträglich in die Fachfeinkonzeption einarbeiten zu können. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).
	Abnahme / Beurteilung Technisches Konzept **(QG 3.1)**	Die Technische Konzeption wird unter Umständen von einem technisch versierten Anwendungsberater, externen Anwendungsentwickler oder unerfahrenen internen Junior-Anwendungsentwickler durchgeführt. In jedem Fall, und speziell in den genannten Fällen, ist eine Abnahme des Technischen Konzeptes durch einen erfahrenen Anwendungsentwickler (Qualitätsmanager) erforderlich. Dieses Quality Gate ist sehr wichtig, da hier die letzte Chance besteht im Vorfeld Einfluss auf die Anwendungsentwicklung zu nehmen und damit teure technische Irrwege zu vermeiden. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).

	Mitwirkung Erstellung Testfallbeschreibung	Jede Anwendungsentwicklung ist intensiv zu testen. Diese Tests beginnen mit kleinen Komponententests (einzelne Funktionalitäten), gehen über die Funktionstest (Test der gesamten Funktion bzw. des gesamten Prozesses), Integrationstest (Test des Zusammenspiels mit allen anderen Funktionen / Prozessen), Regressionstests (Test von theoretisch nicht beeinflussten Funktionen / Prozessen) bis hin zu Last- und Performancetests (Laufzeitverhalten bei einem einzelne User und unter Systemlast). Hierfür werden detaillierte, wiederholbare, vollständige und konsistente Testfall-beschreibungen benötigt. Diese Testfall-beschreibungen sind vom Anwender / Kunden zum Zeitpunkt der Anforderungsformulierung zu erstellen, da es zu jeder Anforderung, zur Verifizierung deren Vollständigkeit und Korrektheit, geeignete Testfälle geben muss. Eine teils erhebliche Detaillierung der Testfälle erfolgt durch die Anwendungsberatung im Rahmen der Fachfeinkonzeption und durch die Anwendungsentwicklung im Rahmen der technischen Konzeption. Die Mitwirkung der Anwendungsentwicklung an der Testfall-beschreibungserstellung kann sehr intensiv und wichtig sein, da bei vielen Entwicklungen Anwender und Anwendungsberatung gar nicht alle theoretisch / technisch möglichen Anwendungsfälle (er-)kennen. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).

	Abnahme / Beurteilung Testfallbeschreibung **(QG 3.2)**	Sind die Testfallbeschreibungen vollständig und konsistent? Passen Sie zur Anforderung und zur geplanten Umsetzung? Diese und weitere Fragen sind durch einen erfahrenen Anwendungsentwickler (bzw. Qualitätsmanager) beim Durchlaufen dieses Quality Gates zu beantworten. Hier gilt erneut: Je weniger erfahrene interne Anwendungsentwickler an der technischen Konzeption beteiligt waren, umso wichtiger ist eine gewissenhafte Abnahme der Testfallbeschreibungen. Zudem offenbart ein Abgleich der Anforderungen mit den Testfall-beschreibungen und ein Abgleich der Testfall-beschreibungen mit der Technischen Konzeption ggf. existierende Inkompatibilitäten von Anforderung und technischer Konzeption und erlaubt somit eine Falsifikation der Spezifikation. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Blueprint (BBP).

5.3 Implementierung (Build)

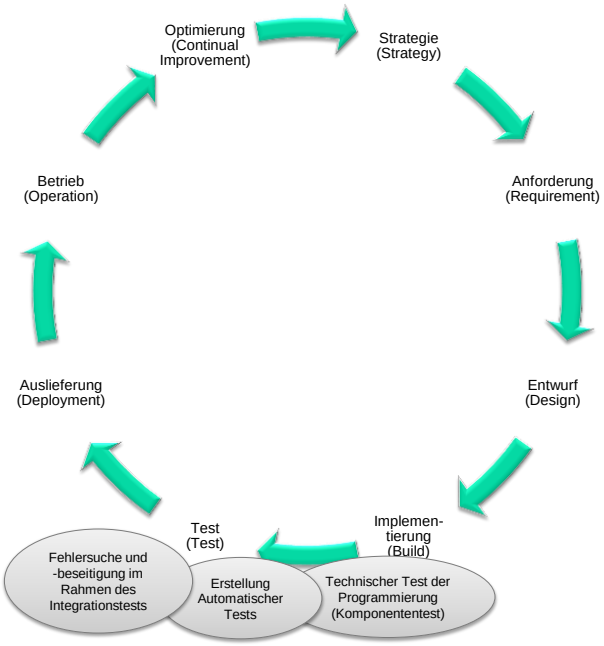


Phase	Aktivität	Beschreibung
Implementierung	**Programmierung**	Auf Basis der technischen Konzeption ist die Anwendungsentwicklung durch einen oder mehrere entsprechend fähige Anwendungsprogrammierer umzusetzen. Die Steuerung der Gesamtheit der Änderungen im Sinne eines ITIL-konformen Change- und Release Managements erfolgt üblicherweise über einen unternehmensweiten Änderungsmanagementprozess. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Quality Gate Management (QGM), SAP Solution Manager Business Process Configuration (BPC), ABAP Development Workbench (DWB), Customizing Comparison and Distribution (CCD / CSCOUT).
	Begleitung / Coaching der Programmierung	Sollte die Umsetzung durch unerfahrene oder externe Anwendungsprogrammierer oder sogar im Offshoring-Szenario erfolgen, so ist die Begleitung / das Coaching durch einen erfahrenen internen Anwendungsentwickler (und Anwendungsberater) sinnvoll und zwingend notwendig.

	Technische Abnahme der Programmierung **(QG 4.1)**	Das Ergebnis der Programmierung ist explizit abzunehmen. Zum einen ist durch einen erfahrenen Anwendungsentwickler (bzw. Qualitätsmanager) zu prüfen, ob die entwickelte Lösung im Bezug auf das technische Konzept vollständig (Wurde alles umgesetzt?) und konsistent (Wurden keine offensichtlichen Fehler, Sicherheitslücken, Programmiersünden oder "Viren" eingebaut?) ist. Zum anderen ist ein Abgleich mit den verbindlich festgelegten Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation durchzuführen. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Pretty Printer (PP), SAP Syntax Check (SC), SAP Extended Syntax Check (ESC), SAP Unicode Checker (UC), SAP Code Inspector (CI), SAP Modification Browser (MB) / SAP Solution Manager Modification Overview (MO), SAP Solution Manager Clone Finder (CF), SAP Solution Manager Interface and Dynamic Call Analysis (IA), SAP Solution Manager Custom Code Remote Compare (CCRC), SAP Solution Manager Change and Transport System Analysis (CTSA), SAP Solution Managers Business Process Change Analyzer (BPCA), SAP Software Delivery Composer (SSDC), SAP Solution Manager Configuration Validation (CV).

	Entwicklungs-koordination	Anwendungsentwickler und -programmierer besitzen technische und theoretische Schwerpunkte, sind jedoch fachliche Allrounder und somit in Projekten gleichzeitig für verschiedene Aufgabenpakete / Teilprojekte einsetzbar. Bei Großprojekten ist es daher aus Effizienz- und Effektivitätsgründen sinnvoll, einen "Entwicklerpool" zu etablieren, aus dem sich die Aufgabenpakete bzw. Teilprojekte kontrolliert bedienen können. Die Steuerung / Verwaltung dieses Pools obliegt einem erfahrenen Anwendungsentwickler, der gleichzeitig für das Coaching und die Qualitätssicherung verantwortlich und zuständig sein kann. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Quality Gate Management (QGM).

5.4 Test (Test)

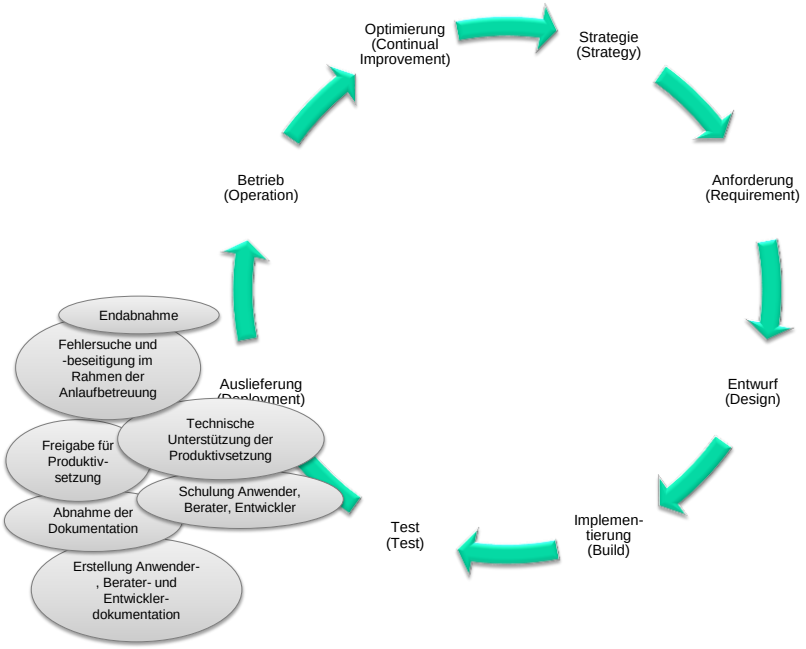

Optimierung
(Continual
Improvement)
Strategie
(Strategy)
Anforderung
(Requirement)
Entwurf
(Design)
Implemen-
tierung
(Build)
Technischer Test der
Programmierung
(Komponententest)
Test
(Test)
Erstellung
Automatischer
Tests
Fehlersuche und
-beseitigung im
Rahmen des
Integrationstests
Auslieferung
(Deployment)
Betrieb
(Operation)

Phase	Aktivität	Beschreibung
Test	**Technischer Test der Programmierung (Komponenten-test)**	Die Anwendungsentwicklung ist für die erste Teststufe verantwortlich und zuständig: Jeder Anwendungsentwickler hat selbständig seine Entwicklungen in geeigneter Form zu testen. Besteht die Anwendungsentwicklung aus vielen einzelnen seiteneffektfreien Funktionalitäten (z.B. Funktionsbausteine, Unterprogramme, Methoden), so können diese durch direkte Ausführung (mit Übergabe geeigneter Eingabewerte und Beurteilung der Ergebniswerte) verifiziert werden. Ist ein solcher Test aufgrund der Architektur bzw. der Abhängigkeiten nicht möglich, so kann auch die Gesamtfunktion ausgeführt und mit den Debugging-Werkzeugen beobachtet bzw. testweise manipuliert werden. Für eine reibungsfreie Implementierung von Änderungen an einer Anwendungslösung wird mindestens eine dreistufige Systemlandschaft bestehend aus Entwicklungs-, Test- und Produktionssystem benötigt. Der Test hat, bis auf wenige begründete Ausnahmen, im Entwicklungssystem und nicht im Testsystem stattzufinden. Der Test findet üblicherweise vor der technischen Abnahme (QG 4.1) statt, damit der Testbericht mit in die Abnahme einfließen kann. Werkzeuge: ABAP Development Workbench (DWB), SAP Test Environment (TE), ABAP Debugger (DBG).

	Erstellung automatischer Tests	Tests sind notwendig, jedoch größtenteils aufwändig, "langweilig", fehleranfällig und personalkostenintensiv. Vor allem mit Blick auf die häufige Wiederholung der Tests innerhalb eines Projektes durch mehrere Testzyklen und Testarten und später bei jeglichen Upgrades (Hinweise, Support Packages, Enhancement Packages, Releasewechsel), ist eine maximale Automatisierung sinnvoll und finanziell lukrativ. Während der Anwendungsberater im Rahmen der Testautomatisierung den Test ganzer Funktionen bzw. Prozesse automatisieren wird - was aber meist aufgrund der größeren Abhängigkeiten und Variationen schwer vollständig möglich ist -, kann und sollte der Anwendungsentwickler den Komponententest weitestgehend automatisieren. Neben dem halbautomatischen Test, durch Hinterlegung von Eingabewerte für Funktionsbausteine und Methoden im Testverzeichnis der Testumgebung, ist eine Ausprägung von Unittests möglich und für geeignete Funktionalitäten sehr sinnvoll. Hierbei handelt es sich um Blackboxtest von gekapselten, seiteneffektfreien Funktionen durch Aufruf der Funktionen mit geeigneten, fest definierten Eingabewerten und Vergleich der tatsächlichen Ergebnisse mit den erwarteten Ergebniswerten. Ergänzend ist eine automatische Laufzeit- und Speicher-verbrauchsbeurteilung möglich. Werkzeuge: SAP Test Environment (TE) + SAP Test Data Directory (TDD), SAP Unit Test Framework (UTF), SAP extended Computer Aided Test Tool (eCATT), SAP End-User Experience Monitoring (EEM).

	Fehlersuche und -beseitigung im Rahmen des Integrationstests	Mit der Übergabe der implementierten Anwendungslösung an den Anwendungsberater zum Funktionstest, ist die Arbeit für den Anwendungsentwickler vorerst beendet. Sollten im Rahmen des Funktionstests, Integrationstests, Regressionstests bzw. Last- und Performancetests jedoch Fehler an der Anwendungslösung identifiziert werden, so sind diese durch die Anwendungsentwicklung schnellstmöglich zu analysieren und zu beheben. Die Fehlerbearbeitung inkl. Fehlermeldung vom bzw. Rückmeldung an den Tester und die Fehlerbehebung durch den Anwendungsentwickler erfolgen mittels eines kontrollierten Fehlermanagement- und Änderungsmanagementprozesses. Die Fehlersuche selbst wird durch zahlreiche Debugging-, Tracing-, Logging- und Monitoring-Werkzeuge unterstützt. Sollte das Fehlverhalten im Entwicklungssystem reproduzierbar sein, so sind die Technische Abnahme und der Funktionstest zu hinterfragen bzw. zu wiederholen. Werkzeuge: SAP Solution Manager Service Desk (SD), SAP Solution Manager Root Cause Analysis (RCA), SAP Solution Manager Change Request Management (ChaRM), ABAP Development Workbench (DWB).

5.5 Auslieferung (Deployment)



Phase	Aktivität	Beschreibung
Auslieferung	**Erstellung Anwender-, Berater- und Entwickler-dokumentation**	Für die Übernahme in den Betrieb ist eine vollständige und konsistente Dokumentation der Änderung erforderlich. Die Entwicklungs-dokumentation ist im Idealfall identisch mit dem Technischen Konzept und ist vom Anwendungs-entwickler zu erstellen. Je nach Projekt-gegebenheiten erfolgt die Erstellung zeitgleich mit der technischen Konzeption oder während der Programmierung oder spätestens nach erfolgreichem Test, wobei letzteres aufgrund des ggf. größeren zeitlichen Abstandes nicht als die zu favorisierende Variante anzusehen ist. Die Beraterdokumentation entspricht dem von der Anwendungsberatung erstellten Technischen Konzept und die Anwenderdokumentation dem vom Anwendungsberater erstellten Fachfein-konzept. Bei beiden kann die Anwendungs-entwicklung unter Umständen mitwirken, ist jedoch nicht verantwortlich und meist nicht zuständig. Zu beachten ist in diesem Zusammen-hang dass es neben dieser, einzig auf die

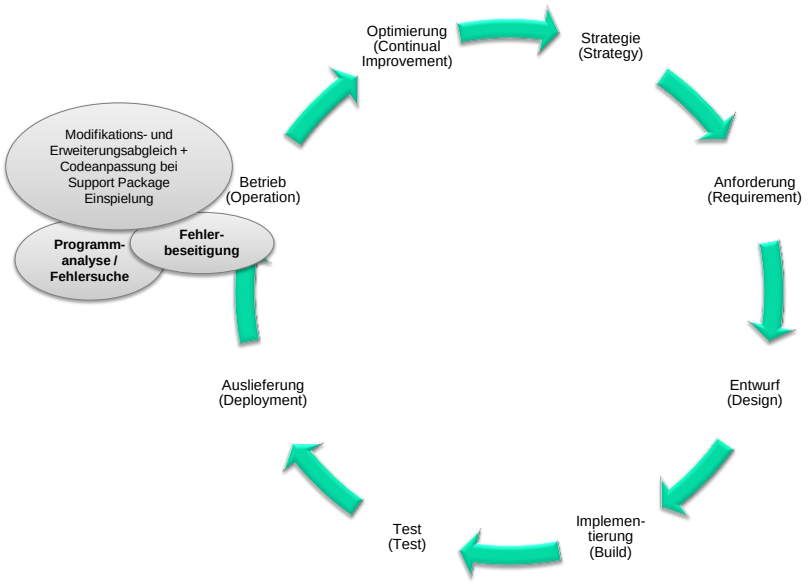
		Änderung bezogene, Deltadokumentation, welche direkt am Änderungsvorgang abgelegt wird, auch eine Gesamtlösungsdokumentation gibt, welche stets die aktuell produktive Gesamtlösung aus prozessualer, fachlicher und technischer Sicht beschreibt, als wichtiger Bestandteil der IT Service Management Configuration Management Database (ITSM CMDB) anzusehen ist, nicht einfach nur eine Sammlung einzelner teilweise ggf. bereits wieder veralteter Change Requests darstellt, und im Rahmen dieser Aktivität ebenfalls anzupassen ist. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Process Repository (BPR), SAP Solution Manager Business Process Configuration (BPC), SAP Solution Manager Solution Database (SolDB), SAP Solution Manager Custom Code Lifecycle management Database (CCDB), SAP Solution Manager eLearning Management (ELM).
	Abnahme der Dokumentation **(QG 4.2)**	Wie bei vielen vorangehenden Aktivitäten gilt: Da die Tätigkeit der Dokumentationserstellung unter Umständen von unerfahrenen internen oder erfahrenen externen Anwendungsentwicklern durchgeführt wurde und generell ein Vier-Augen-Prinzip sinnvoll ist, ist eine explizite Abnahme der Entwicklerdokumentation erforderlich. In der Regel erfolgt diese Abnahme im Rahmen der technischen Abnahme der Programmierung, spätestens jedoch im Zuge der Auslieferung. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Process Configuration (BPC), SAP Solution Manager Solution Database (SolDB), SAP Solution Manager Custom Code Lifecycle Management Database (CCDB), SAP Solution Manager Solution Documentation Assistant (SoDocA).
	Schulung Anwender,	In den meisten Fällen wird der für die Konzeption

	Berater, Anwendungs-entwickler	und Umsetzung zuständige Anwendungs-entwickler nicht für den Betrieb, die Fehler-beseitigung und für kleinere Weiter-entwicklungen zuständig sein. Es ist somit eine geordnete Übergabe an die Anwendungs-entwicklung der Anwendungsbetreuung erforderlich. Die Übergabe der Entwickler-dokumentation ist hierbei ein wichtiger Bestandteil. Notwendig ist jedoch zusätzlich die persönliche Einweisung des übernehmenden Anwendungsentwicklers in die Lösung. Auch kann es sinnvoll sein, dass der ausführende Anwendungsentwickler bei Berater- (bzw. Betreuer-) und Anwenderschulungen involviert ist um technische Spezialfragen schnell klären zu können. Werkzeug: SAP Solution Manager eLearning Management (ELM).
	Freigabe für Produktivsetzung **(QG 5)**	Die Anwendungsentwicklung, die anschließend auch den Betrieb verantwortet, ist im Sinne der Qualitätssicherung aktiv in die sehr wichtige Release Management Phase "Freigabe der Änderung für die Produktion" eingebunden. Hierbei wird, durch den Release Manager und durch weitere Entscheidungsträger (inkl. Kunde und Anwendungsentwicklung), vollumfassend auf Basis der vorangehenden Bewertungen / Beurteilungen / Abnahmen geprüft, ob alle Voraussetzungen für eine erfolgreiche Produktiv-setzung und einen anschließenden stabilen Betrieb gegeben sind oder ob noch an der einen oder anderen Stelle Nachbesserungen und eine damit verbundene Terminverschiebung einzelner Änderungen oder des gesamten Releases notwendig wird. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Quality Gate Management (QGM).
	Technische Unterstützung der Produktivsetzung	Die Produktivsetzung einer großen Änderung ist meist mit zahlreichen organisatorischen, fachlichen und technischen Vor- und Nacharbeiten verbunden, welche ausführlich in

		einem projektspezifischen CutOver-Plan dokumentiert sind. Der Import der Änderung in das Produktivsystem ist hierbei im Prinzip eine einzelne unkritische Aktivität. Sollten die Tests jedoch unvollständig durchgeführt wurden sein, der Entwicklungsstand des Produktionssystems signifikant vom Testsystem abweichen oder das Testsystem noch Entwicklungen (des gleichen oder eines anderen Projektes) enthalten, die noch nicht produktiv gehen sollen, dann kann es zu teils gravierenden technischen Problemen (z.B. Programmsyntaxfehler) beim Import kommen. Diese Probleme sind dann durch einen auf Abruf bereitstehenden erfahrenen Anwendungsentwickler zeitnah zu beheben. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Enhanced Change and Transport System (CTS+), SAP Software Delivery Assembler (SSDA).
	Fehlersuche und -beseitigung im Rahmen der Anlaufbetreuung	Der Quotient aus der Anzahl der Störungs- und Änderungsanforderungsmeldungen der Anlaufbetreuung und der Anzahl der Fehlermeldungen der Testphase, ermöglicht einen direkten Rückschluss auf die Qualität der Phasen "Anforderung" (Wurde etwas vergessen?) und "Test" (Wurde etwas nicht getestet?) und somit auf Vollständigkeit und Korrektheit der Ergebnisse des Implementierungsprojektes. Unabhängig von dieser Qualität treten direkt nach der Produktivsetzung in der Regel die meisten und kritischsten Störungen zu tage, so dass hier die Fehlersuche und -beseitigung u.U. in ähnlicher Stärke wie beim Integrationstest abzusichern ist. Das Vorgehen ist analog, jedoch wird statt mit Fehlermeldungen nun mit "Incidents" und "Problems" zur Störungs- und Problemmeldung bzw. -behebung gearbeitet. Störungen bzw. Probleme und deren Workarounds bzw. Lösungen werden in einer Wissensdatenbank dokumentiert, um bei einem erneuten Auftreten die Behebung zu beschleunigen bzw. durch den First Level Support zu ermöglichen.

		Werkzeuge: SAP Solution Manager Service Desk (SD), SAP Solution Manager Root Cause Analysis (RCA), SAP Solution Manager Change Request Management (ChaRM), ABAP Development Workbench (DWB).
	Endabnahme **(QG 6)**	Die Endabnahme ist ein formaler Akt zwischen Projektleitung und Kunde und definiert das offizielle Ende des Projektes inkl. Auflösung des Projektteams, inkl. Rechnungslegung, inkl. vollständige Übernahme des Betriebs durch die Anwendungsbetreuung (und deren Anwendungsentwicklung). Die Anwendungs-entwicklung ist hier nur am Rande als Bestandteil des aufzulösenden Projektteams involviert, kann jedoch unter Umständen für vereinbarte Nacharbeiten / Folgeaktivitäten verantwortlich und zuständig sein. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Quality Gate Management (QGM).

5.6 Betrieb (Operation)

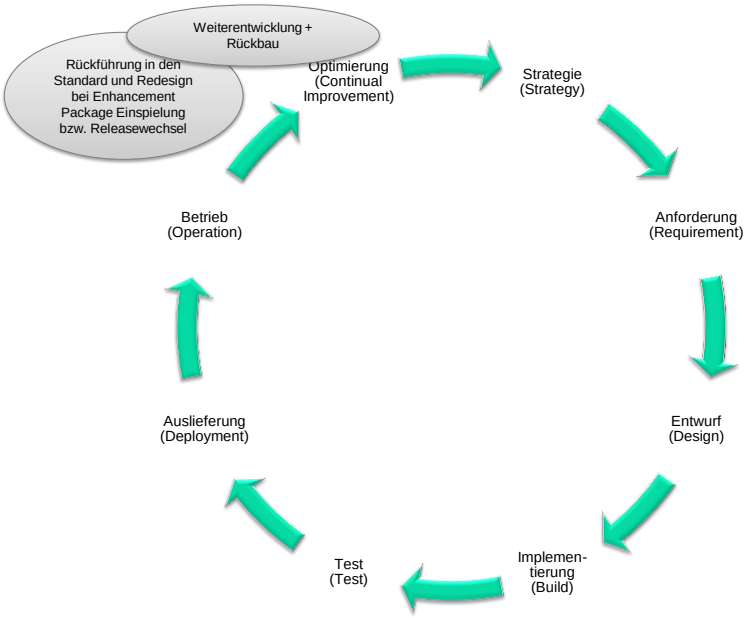


Phase	Aktivität	Beschreibung
Betrieb	**Programmanalyse / Fehlersuche**	Eine größere Softwarelösung ist komplex und somit niemals fehlerfrei. Daher werden auch nach erfolgreicher Anlaufbetreuung noch "merkwürdige" Phänomene festgestellt werden, die dann zeitnah zu analysieren und im Falle eines Fehlers schnellstmöglich zu beheben sind. Die Fehlerbearbeitung inkl. Fehlermeldung vom bzw. Rückmeldung an den Meldenden und die Fehlerbehebung durch den Anwendungs-entwickler erfolgen mittels eines kontrollierten Störungs-, Problem- und Änderungs-managementprozesses. Durch Einsatz geeigneter Werkzeuge ist zusätzlich eine proaktive Störungserkennung möglich, was für den Endanwender eine signifikante Verbesserung der gefühlten Lösungsqualität zur Folge hat und was der Anwendungsbetreuung die schnelle Behebung der Störung ermöglicht, bevor teure Folgeschäden und -kosten entstehen bzw. bevor der Endanwender die Störung überhaupt bemerkt. Werkzeuge: SAP Solution Manager Service Desk (SD), SAP Solution Manager Root Cause Analysis (RCA), SAP Solution Manager Configuration Validation (CV), SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Business Process Monitoring (BPMon), SAP Solution Manager End-User Experience Monitoring (EEM).

	Fehlerbeseitigung	Die Anwendungsentwicklung ist für die kontinuierliche Wartung der Gesamtlösung zuständig, wobei einzelne Fehler unter Umständen im Rahmen der Gewährleistung und Anwendungsbetreuung an Dritte (z.B. SAP) weitergereicht werden können. Die Qualitätssicherung der vorangehenden Beurteilungs- und Abnahmeschritte macht sich nun bei den Aktivitäten Fehlersuche, Fehlerbeseitigung und Weiterentwicklung bezahlt, da im Rahmen der Qualitätssicherung zu prüfende bzw. zu sichernde Eigenschaften wie Reife, Fehlertoleranz, Komplexität, Stabilität, Prüfbarkeit, Analysierbarkeit, Modifizierbarkeit, Erweiterbarkeit, ... einen direkten und signifikanten Einfluss auf den Zeit-, Budget- und Ressourcenbedarf dieser Aktivitäten haben. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), ABAP Development Workbench (DWB).

	Modifikations- und Erweiterungsabgleich + Codeanpassung bei Support Package Einspielung	Die kundeneigenen Änderungen basieren auf der Standardlösung "SAP Business Suite". Diese Standardlösung ist nicht frei von Fehlern, so dass auch hier kontinuierliche Fehlerbeseitigungen durch Einspielung von durch SAP bereitgestellten Hinweisen, durch Einspielung von SAP Support Packages oder durch Ausführung manueller Korrekturaktivitäten per Modifikation notwendig sind. Diese Änderungen an der Standardlösung sind durch die Anwendungsentwicklung durchzuführen und haben unter Umständen Auswirkungen auf kundeneigene Entwicklungen, welche durch die Anwendungsentwicklung zu begutachten und zu korrigieren sind: Es ist in diesem Zusammenhang ein Modifikations- und Erweiterungsabgleich durchzuführen. Außerdem ist eine Änderungseinflussanalyse durchzuführen, um Anpassungsbedarfe aufgrund der Verwendung veränderter Standardobjekte in kundeneigenen Entwicklungen zu identifizieren. Zudem ist die Durchführung von Syntaxprüfungen für alle kundeneigene Programme, Funktionsgruppen und Klassen und das Aufspüren von kundeneigenen Entwicklungen, die ursprünglich als Kopien von SAP Standardentwicklungen angelegt wurden und nun unter Umständen analog zum Standard anzupassen sind, sinnvoll bzw. notwendig. Werkzeuge: SAP Note Assistant (SNOTE), SAP Solution Manager Maintenance Optimizer (MOPZ), SAP Support Package Manager (SPAM), SAP Add-on Installation Tool (SAINT), SAP Modification & Enhancement Toolset (MET), SAP Solution Manager Upgrade / Change Impact Analysis (UCIA), SAP Unicode Checker (UC), SAP Solution Manager Clone Finder (CF).

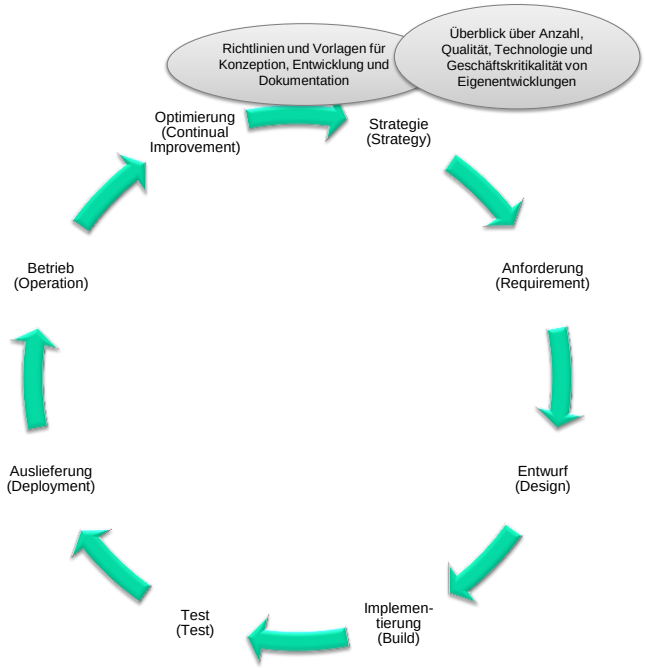
5.7 Optimierung (Continual/Continuous Improvement)



Phase	Aktivität	Beschreibung
Optimierung	**Rückführung in den Standard und Redesign bei Enhancement Package Einspielung bzw. Releasewechsel**	Die SAP Business Suite wird von SAP kontinuierlich weiterentwickelt und sukzessiv mittels Durchführung eines Releasewechsels oder mittels Einspielung von Enhancement Packages mit anschließender Aktivierung gewünschter Business Functions um neue Funktionalitäten angereichert. Hier sind durch die Anwendungsentwicklung im Prinzip die gleichen Aktivitäten wie bei einer Fehlerkorrektur an der Standardlösung durchzuführen. Zusätzlich kann es nun jedoch vorkommen, dass eine kunden-eigene Lösung auf Standardobjekten basiert, die sich im Rahmen des Upgrades grundlegend in Ihrer Schnittstelle oder Ihrem Verhalten geändert haben oder von SAP nicht mehr weiterentwickelt / supportet werden. In diesen Fällen ist unter Umständen ein komplettes Redesign der kundeneigenen Lösung notwendig. Auch kann es sein, dass die durch die kundeneigene Anwendungsentwicklung bereitgestellte Funktionalität nun im Standard verfügbar ist und somit in den Standard zurückgeführt werden kann / sollte. Werkzeuge: SAP Solution Manager Maintenance Optimizer (MOPZ), SAP Upgrade Dependency Analyzer (UDA), SAP Software Logistic Toolset (SLT), SAP Modification & Enhancement Toolset (MET), SAP Solution Manager Upgrade/Change Impact Analysis (UCIA), SAP Unicode Checker (UC), SAP Solution Manager Clone Finder (CF), Enhancement & Switch Framework (SF), ABAP Development Workbench (DWB).

	Weiterentwicklung + Rückbau	Kundeneigene Lösungen werden durch die Anwendungsentwicklung auf Basis neuer oder veränderter Kundenanforderungen kontinuierlich weiterentwickelt. Die Anwendungsentwicklung ist hier zuständig, während die Anwendungs-beratung oder -betreuung verantwortlich ist. Zahlreiche kundeneigene Entwicklungen besitzen nur eine kurze Lebensspanne, da die Lösungen manchmal (z.B. im Rahmen eines Migrationsprojektes) nur einmal benötigt werden oder irgendwann durch den Standard oder eine neue Version abgelöst werden. Die veralteten nicht mehr benötigten kundeneigenen Entwicklungen sind dann zurückzubauen, damit diese in zukünftigen Upgrades nicht mehr berücksichtigt werden müssen bzw. damit von diesen keine Gefahr mehr für den stabilen Betrieb der Gesamtlösung ausgeht. Werkzeuge: SAP Solution Manager Clearing / Using Analysis (CA), ABAP Development Workbench (DWB).

5.8 Strategie (Strategy)



Phase	Aktivität	Beschreibung
Strategie	**Richtlinien und Vorlagen für Konzeption, Anwendungs-entwicklung und Dokumentation**	Für die Entwicklung von Anwendungen gelten Regeln, welche teilweise als Empfehlungen anzusehen und teilweise verpflichtend einzuhalten sind. Hierfür existieren u.a. die unternehmenspezifischen, an öffentlichen Publikationen orientierten, Richtlinien und Vorlagen für Konzeption, Anwendungs-entwicklung und Dokumentation. Diese Richtlinien und Vorlagen werden von der Anwendungsentwicklung erstellt und regelmäßig aktualisiert. Sie sind von sämtlichen (internen und externen) Anwendungsentwicklern einzuhalten bzw. zu verwenden. Die Einhaltung / Verwendung wird in entsprechenden Quality Gate überprüft bzw. sichergestellt. Die Richtlinien und Vorlagen sind offizieller Bestandteil des Qualitätsmanagement-regelwerkes des beratenden, entwickelnden bzw. betreuenden Unternehmens. Werkzeuge: SAP Solution Manager Change Request Management (ChaRM), SAP Solution Manager Quality Gate Management (QGM).

	Überblick über Anzahl, Qualität, Technologie und Geschäfts-kritikalität von Eigenentwicklungen	Das Entwickeln von Anwendungen ist kostenintensiv und risikohaft. Den Überblick zu behalten, ist daher die wichtigste Aufgabe der Anwendungsentwicklung. Andernfalls kommt es aufgrund des nicht mehr beherrschbaren Risikos und des immer größeren Pflegeaufwandes zum Stillstand, da sämtliche Kräfte für das Am-Leben-Erhalten statt für Innovation eingesetzt werden und irgendwann kommt es dann zu einem Kollaps, da nicht einmal das Am-Leben-Erhalten mehr möglich ist. Es ist daher für die Anwendungsentwicklung bzw. für ein Unternehmen sehr wichtig, stets folgende Fragen beantworten zu können: ▪ Welche kundeneigenen Entwicklungen gibt es? Welche davon werden wie häufig verwendet? Wieviel sind es absolut und relativ zum Standard? In welchen Systemen befinden sich welche Versionen der Entwicklungen? ▪ Wie hoch sind die Qualität, das Risiko, die Effizienz und die Effektivität der einzelnen Entwicklungen? (Korrektheit, Vollständigkeit, Sicherheit, Reife, Fehlertoleranz, Wieder-herstellbarkeit, Verständlichkeit, Erlernbarkeit, Bedienbarkeit, Ergonomie, Zeitverhalten, Verbrauchsverhalten, Unabhängigkeit, Komplexität, Genauigkeit, Analysierbarkeit, Modifizierbarkeit, Erweiterbarkeit, Stabilität, Prüfbarkeit, Installierbarkeit, Austausch-barkeit, ...) ▪ Welche Technologie wurde eingesetzt / angewendet? Handelt es sich um die bestmögliche Lösung? Ist sie zukunftssicher, stabil, wartbar, erweiterbar? (Modifikation mit oder ohne Assistent, Erweiterung mit oder ohne Interface, Eigenentwicklung mit oder ohne Verwendung nicht-freigegebener Standardfunktionen, SAPscript / Smartform / AdobeForm, ABAP Report / ABAP Dynpro / WebDynpro Java / Visual Composer / WebDynpro ABAP / Floorplan Manager / Business Server Pages / SAP Webclient UI, SAP Workflow / Business Process Management / ...)

	▪ In welchen Geschäftsprozessen wird die Anwendungsentwicklung verwendet? Häufig genutzte wichtige Kerngeschäftsprozesse oder selten genutzte Nebenprozesse? Gibt es im Falle einer Störung der Anwendungs-entwicklung einen Workaround zur Umgehung der Störung, einen Fallback-Plan zur Rückkehr auf einen funktionierenden Stand oder einen Backout-Plan zur Rückgängig-machung der letzten Änderung? Wie schnell ist eine Wiederherstellung der Funktionalität möglich? Werkzeuge: SAP Solution Manager Solution Database (SolDB), SAP Solution Manager Custom Code Lifecycle Management Database (CCDB).

6 Vertiefungsrichtungen und Erfolgsfaktoren

Sehr geehrte Leserin, sehr geehrter Leser,

dieses Handbuch nähert sich seinem Ende...und dabei gäbe es noch so viel zu sagen. Zum Beispiel zur konkreten, in diesem Handbuch lediglich angerissenen und keinesfalls vollständigen, Ausgestaltung der IT Service Management Assets (inkl. Prozesse) im Kontext Custom Code Lifecycle Management [1]. Und ganz konkret zum Change- und Release Management [2] als den für die Anwendungsentwicklung wichtigsten IT Service Management Prozess, zum Upgrade Management [3] als die für die Anwendungsentwicklung wichtigste und kritischste Aktivität und zum Template Management [4] also eine der wichtigsten und interessantesten Spezialisierungen des Custom Code Lifecycle Managements. Und zur Anwendung der zahlreichen Werkzeugen der ABAP Development Workbench und des SAP Solution Managers [5] als die wichtigsten Hilfsmittel des Anwendungsentwicklers. Und zum Add-on Assembly Kit und dem Software Logistic Toolset der SAP SE [6] zur Erstellung und Auslieferung von ABAP Add-ons. Und zu konkreten Vorschlägen für Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation [7] als Sammlung von bewährten Best Practices für eine "gute" Anwendungsentwicklung. Und zum Aufbau einer internen + externen Entwicklungsberatung [8] zur Einführung / Optimierung / Durchführung eines aktiven Custom Code Managements. Und und und.

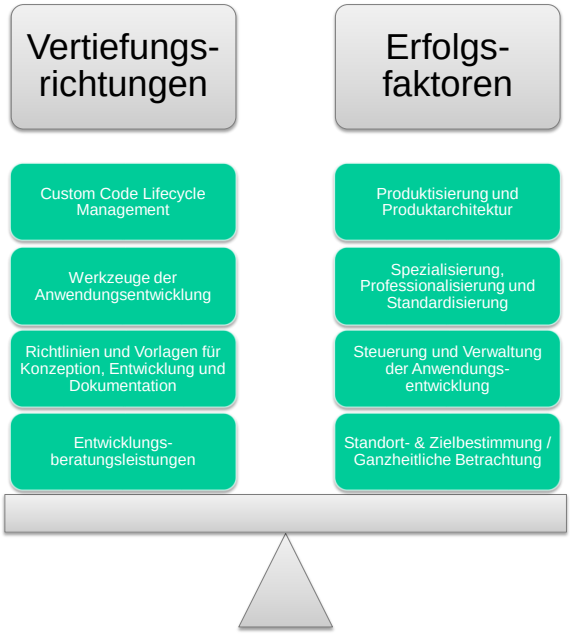
Dieses Handbuch soll Ihnen als Ausgangpunkt dienen. Es soll der Anstoß sein, um sich kritisch, konstruktiv und zielorientiert mit der Situation und der Zukunft der eigenen Anwendungsentwicklungsabteilung auseinander zu setzen. Es soll Ihnen Wege + Ziele aufzeigen und wichtige Impulse auslösen. Aber: Es ist nur ein Baustein auf dem Weg zum Erfolg und nur ein Teil eines größeren Ganzen: Die Anwendungsentwicklung benötigt eine Struktur, welche durch die Produktisierung / Modularisierung und die verschiedenen sich ergänzenden Softwarearchitekturmodelle (Serviceorientierte Architektur + Client-Server-Architektur + Model-View-Controller-Entwurfsmuster) gegeben ist [A]. Die Anwendungsentwicklung benötigt Input von und liefert Output an eine spezialisierte, professionalisierte und standardisierte Anwendungsberatung und -betreuung [B]. Die Anwendungsentwicklung bedarf einer Verwaltung und Steuerung in der Linie (IT Service Management / Custom Code Lifecycle Management) und im Einsatz (Projektmanagement + Systems/Software Engineering) [C]. Und: Die Anwendungsentwicklung bedarf einer vorangehende Standort- und Zielbestimmung. Zudem ist die Anwendungsentwicklung ein sehr stark vernetzter Teil Ihres Unternehmens und kann daher nur ganzheitlich (zusammen mit allen anderen verbundenen Abteilungen und korrelierenden Aspekten) betrachtet und optimiert werden. [D].

Sie sehen: Es ist viel zu tun...aber: es lohnt sich...also: packen Sie's an!

Ich wünsche Ihnen viel Erfolg!

Ihr

Peter Weigel



6.1 Custom Code Lifecycle Management

Dieses Handbuch dient dem Einstieg in das Thema Custom Code Lifecycle Management. Einige Themen wurden jedoch gar nicht oder nur oberflächlich behandelt. Einige Dienstleistungsbestandteile wurden z.B. überhaupt nicht angesprochen. Gleiches gilt für die meisten IT Service Management Prozesse. Und selbst die Themen, die detaillierter diskutiert wurden, sind auf das eigene Unternehmen zu übertragen, anzupassen und zu ergänzen. Hierfür sind zwingend weitere Publikationen sowie das unternehmenseigene Wissen einzubeziehen.

Zum Thema Change-, Release- und Transport Management sei auf den SAP Change- & Release Management Implementation Guide verwiesen. Zum Thema Upgrade Management sei auf den SAP Upgrade Management Guide verwiesen. Zum Thema Template Management sei auf andere Literaturquellen verwiesen.

6.2 Werkzeuge der Anwendungsentwicklung

Die unterstützenden Werkzeuge der ABAP Development Workbench und des SAP Solution Managers wurden in Abschnitt 4.4 kurz vorgestellt. Für eine ausführliche Betrachtung sei auf entsprechende Publikationen und Anwendungsdokumentationen verwiesen.

Auch die unterstützenden Werkzeuge im Umfeld ABAP Add-on Assembly Kit und SAP Software Logistic Toolset wurden in Abschnitt 4.4 kurz vorgestellt. Hier sei ebenfalls für

eine ausführliche Betrachtung auf entsprechende Publikationen und Anwendungs-dokumentationen verwiesen. Als zentrale Informationsplattform zum Thema Add-on Anwendungsentwicklung dient das SAP Integration & Certification Center (http://www.sap.com/icc).

6.3 Richtlinien und Vorlagen für Konzeption, Entwicklung und Dokumentation

Die Programmiersprache ABAP zu beherrschen und lauffähige, korrekte, vollständige und genaue Lösungen zu entwickeln ist nur eine Seite der Medaille. Hierfür existieren zahlreiche Programmierhandbücher, die die Programmiersprache ABAP und die zugrundeliegenden allgemeingültigen Theorien und Techniken ausführlich erklären.

Dafür zur sorgen, dass die Lösung sicher, reif, wiederherstellbar, verständlich, erlernbar, bedienbar, ergonomisch, effizient (Zeit + Verbrauch), unabhängig, einfach, analysierbar, modifizierbar, erweiterbar, stabil, prüfbar, installierbar und austauschbar ist, ist die andere (viel wichtigere) Seite. Auch hierfür gibt es (wenn auch deutlich weniger) entsprechende Publikationen, die als Basis für unternehmenseigene Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation dienen können.

6.4 Entwicklungsberatungsleistungen

In Kapitel 5 wurden die zahlreichen Aktivitäten der Anwendungsentwicklung vorgestellt. Diese Auflistung ist mit Sicherheit nicht vollständig. Außerdem gibt es, zu den als Grundlage gewählten IT Service Management Lebenszyklusphasen, alternative Gruppierungen, die für den einen oder anderen Zweck sinnvoller sein können. Zum Beispiel könnten die Aktivitäten "Komponententest", "Debugging", "Tracing", "Logging" und "Monitoring" als verschiedene Möglichkeiten zur "Programmanalyse" verstanden werden. Analog können die Aktivitäten "Konzeptreview", "Coaching" und "Endabnahme" als Aspekte der "Qualitätssicherung" betrachtet werden. Eine derartige Umgruppierung würde einer Orientierung an den konkreten Skills entsprechen, würde den Prozesscharakter verlieren, ließe sich aber besser konkreten Personen zuordnen. Zudem könnte diese Umgruppierung / Clusterung als Grundlage für konkrete Entwicklungsberatungsleistungen dienen:

Cluster	Beschreibung
Konzeption und Programmierung	Es erfolgt die Bereitstellung sehr gut ausgebildeter + praxiserfahrener Anwendungsentwickler- und Programmierer-ressourcen zur Konzeption und Umsetzung konkreter Entwicklungsanforderungen als Kundenentwicklung oder Add-on.
Programmanalyse + Fehlerbeseitigung	Es erfolgt die Bereitstellung sehr gut ausgebildeter + praxiserfahrener Anwendungsentwickler- und Programmierer-ressourcen zur tatkräftigen Unterstützung bei der Programm-analyse und Fehlerbeseitigung im Rahmen eines Integrationstests, einer Anlaufbetreuung oder zur temporären Unterstützung des regulären Betriebs.
Lösungs-architektur und Qualitätssicherung	Es erfolgt die Bereitstellung eines sehr gut ausgebildeten + praxiserfahrenen Lösungsarchitekten zur Durchführung von Coaching-, Koordinations-, Bewertungs-, Beurteilungs- und Qualitätssicherungsaktivitäten.
Wissens-vermittlung	Es erfolgt die Bereitstellung eines sehr gut ausgebildeten + praxiserfahrenen Anwendungsentwicklers zur Durchführung von (interne und externen) Schulungs- und Coachingtätigkeiten mit dem Ziel des Wissenstransfers in einem spezifischen fachlichen oder technologischen Themengebiet.
Upgrade Management	Es erfolgt die professionelle, effiziente, effektive, qualitativ hochwertige und an SAP Best Practices orientierte Durchführung oder Unterstützung eines technischen Upgradeprojektes inkl. routinierter Nutzung der zahlreichen durch die SAP SE im SAP Support Portal und im SAP Solution Manager zur Verfügung gestellten Hilfsmittel / Werkzeuge.

Custom Code Management	Es erfolgt die Durchführung oder Unterstützung von Aktivitäten im Kontext "Aktives Management kundeneigener Entwicklungen": ▪ Aufbau / Aktualisierung einer Custom Code Database ▪ Verknüpfung der Entwicklungen mit der Solution Database (Geschäftsprozesse und deren Dokumentationen) ▪ Erstellung automatischer Tests (Unit Test) ▪ Analyse und Beurteilung der Entwicklungen ▪ Rückführung in den Standard ▪ Rückbau nicht mehr verwendeter / benötigter Entwicklungen, ▪ Verbesserung der Qualität durch Erstellung / Ergänzung der Dokumentationen, durch Migration auf eine geeignetere / releasesicherere Technologie oder durch Reengineering / Redesign der Lösung ▪ Erstellung einer auf Best Practices basierenden unternehmensspezifischen Richtlinien und Vorlagen für Konzeption, Anwendungsentwicklung und Dokumentation ▪ ...
Change-, Release- und Transport Management	Es erfolgt die Durchführung oder Unterstützung eines Projektes zum Aufbau oder zur Optimierung eines "Change-, Release- und Transport Managements von SAP Softwareänderungen mit dem SAP Solution Manager Change Request Management" inkl. Prozessberatung, Transportarchitekturberatung, Anwendungsberatung, Anwendungsentwicklung und kontinuierlicher Anwendungsbetreuung.

Diese Liste der Clusterthemen ist keinesfalls vollständig. Eine Detaillierung ist möglich bzw. nötig. Eine ausführliche Diskussion der konkreten Herangehensweisen zur Kundenakquise für die beiden letztgenannten Themen kann an dieser Stelle nur angedeutet werden und ist im Bedarfsfall entsprechend zu vertiefen:

- Die Anwendungsbetreuung geht auf ihre eigenen Outsourcing-Kunden zu, bietet entsprechende Aktivitäten / Leistungen an und stellt im Gegenzug die Reduktion des Servicepreises für den laufenden Betrieb in Aussicht.
- Die Anwendungsbetreuung führt die entsprechenden Aktivitäten / Leistungen selbständig durch und profitiert in der Folge von einer signifikanten Kostenreduktion und einer damit einhergehenden Renditesteigerung für den laufenden Betrieb.
- Die Anwendungsberatung geht auf externe Kunden zu, bietet entsprechende Aktivitäten / Leistungen an und stellt im Gegenzug die Reduktion des Servicepreises für den laufenden Betrieb in Aussicht. Sollte der IT-Dienstleister des externen Kunden dieser Servicepreisreduktion nicht zustimmen, so besteht die Möglichkeit,

dass die eigene Anwendungsbetreuung dem externen Kunden einen Anwendungs- und Systembetrieb zum reduzierten Servicepreis garantiert und hierdurch den externen Kunden als neuen Outsourcing-Kunden gewinnt.

- Die Anwendungsberatung geht auf andere IT-Dienstleister zu und bietet entsprechende Aktivitäten / Leistungen mit dem Ziel der signifikanten Kostenreduktion und Gewinnsteigerung an. Möglicherweise ergibt sich hieraus eine langjährige, stabile, faire und für beide Seiten fruchtbare Partnerschaft / Zusammenarbeit.

6.5 Produktisierung und Produktarchitektur

Die beiden Themen "Produkt" und "Dienstleistung" sind untrennbar miteinander verwoben. Ein Produkt ist etwas greifbares, das man vervielfältigen, ins Regal stellen und anfassen bzw. ansehen kann (z.B. ein konkretes Add-on-Produkt). Ein Produkt besitzt stets einen Lebenszyklus und wird einem Kunden immer in Kombination mit verschiedenen Dienstleistungen - z.B. einmalige Einführungsberatung und kontinuierliche Betreuung - bzw. mit der Garantie bestimmter Eigenschaften - z.B. Zweckmäßigkeit + Verfügbarkeit - angeboten. Zu jedem Produkt gibt es folglich entsprechende Dienstleistungen (z.B. ein konkreter Add-on Service). Anders herum basiert jede Dienstleistung (z.B. eine konkrete Beratungsleistung) auf einem Produktbündel bestehend aus Software, Hardware, Präsentationen, Angebotsvorlagen, Konzepten und Dokumentationen. Bei genauerer Betrachtung findet sich für jede Dienstleistung ein solches Produkt (z.B. ein konkretes Beratungsprodukt), denn eine Dienstleistung ohne Produktbasis wird niemals dauerhaft existieren bzw. die versprochene Qualität in reproduzierbarer Form liefern können. Es gilt mathematisch

gesprochen: Produkt ≈ Dienstleistung. „Die Serviceorientierung ist eine notwendige aber nicht hinreichende Voraussetzung für die Produktisierung. Die Produktisierung ist eine notwendige aber nicht hinreichende Voraussetzung für die Serviceorientierung."

Das Thema Produktisierung und Produktarchitektur ist aus technisch-rechtlicher Sicht ein sehr komplexes Thema und umfasst folgende Themenkomplexe:

- **Produktarten**: Standardsoftware, Codeschnipsel, Code Template, Code Library / Framework, Add-on, Customizing Template, Application Service, Consulting & Support Service
- **Grundlagen**: Add-on-Definition; Add-on-Eigenschaften; Abgrenzung zu Produkt, Template, Codeschnipsel; Vorteile für Dienstleister und für Kunden
- **Deliverymodell**: Beratungslösung, Wartungs-&Supportvertrag, Freiwilliger kostenloser Support, Bestandteil vom SAP Consulting Add-on Package
- **Stakeholder**: Entwickler (Wie?), Berater (Was?), Führung (Wann?), Kunde (Was zuerst?), Wettbewerb (Was nicht?), Lieferant / SAP (Wie? Was? Was nicht?)
- **Richtlinien und Vorlagen**: Konzeption, Entwicklung, Dokumentation, Qualitätssicherung, Marketing, Angebot, Vertrag, Namensraum (Partnernamensraum, Unternamensraum, Workflownummernbereich, Pseudonamensraum, …), Add-on Assembly Kit, SAP Add-on Certification (ICC)
- **Rechliche Rahmenbedingungen**: Urheberrecht, Nutzungs- und Verwertungsrecht, Inhalte und Rechte Dritter (Kunde, Partner, Mitarbeiter, SAP, OpenSource), Haftung, Garantie, Gewährleistung
- **Nutzungsbedingungen**: Bereitstellung (Technisch), Bereitstellung (Kaufmännisch), Bereitstellung (Sonstiges, z.B. Reparaturlizenz, Referenzkunde), Support (Kontakt Dienstleister -> Kunde), Support (Kontakt Kunde -> Dienstleister), Beratung
- **IT Service Management**: Financial Management, Demand Management, Service Cataloque Management, Change Management, Service Validation & Testing, Release & Deployment Management, Incident Management, …
- **Produkt Lifecycle Management**: Produktentwicklungslandschaft, Produktrelease-management, Template Management, ...
- **Produktarchitektur**: Softwarearchitektur, Systemarchitektur, Transportarchitektur, Servicearchitektur, Prozessstruktur, Organisationsstruktur, Rollenmatrix
- **Vorgehen**: Reifegrad erhöhen, Beratung standardisieren und Best Practices fixieren, Rahmenbedingungen schaffen, Lösungslandschaft aufbauen und Template / Referenzmodell definieren und implementieren, Add-on Suite aufbauen und nutzen

Zusätzlich sind im Zusammenhang mit der Produktentwicklung zahlreiche strategisch-organisatorische Fragestellungen zu klären:

- Product Management Owner: Rechte und Pflichten
- Product Owner: Rechte und Pflichten
- Produkt Lifecycle Management: Absicherung von Beratung, Entwicklung und Betrieb

- Priorität: externes Kundenprojekt vs. interne Produktentwicklung
- Mission + Vision + Strategie + Ziele + Maßnahmen

Generell sind folgende Fragestellungen zum Thema „Welche Rahmenbedingungen sind zu schaffen?“ zu diskutieren:

- Haben wir alle eine klare + einheitliche **Definition** vom Begriff „Add-on“? Welches? Was sind die Eigenschaften, Vor- und Nachteile? Was ist KEIN Add-on?
- Welche **Richtlinien** gelten für die Add-on-Entwicklung? Entwicklung? Dokumentation? Namensraum? Qualität?
- Welche **Werkzeuge / Hilfsmittel** wollen wir nutzen? Add-on Assembly Kit? SAP Add-on Certification?
- Sind alle **rechtlichen Rahmenbedingungen** (Urheberrecht, Nutzungs- und Verwertungsrecht, Inhalte und Rechte Dritter, Haftung, Garantie, Gewährleistung) geklärt und abgesichert? Sind alle Angebote/Verträge/AGBs (Kunde, Partner, externe Berater, Mitarbeiter, SAP, OpenSource) entsprechend geprüft und angepasst?
- Welches **Deliverymodell** (Beratungslösung, Wartungs- & Supportvertrag, Freiwilliger kostenloser Support, Teil des SAP Consulting Add-on Packages) wird genutzt? Wie lauten die konkreten Nutzungsbedingungen (Technisch, Kaufmännisch, Support, Beratung)?
- Sind alle internen **Stakeholder** (Entwicklung, Beratung, Führung, Vertrieb) bekannt und eingebunden? Wer? Wie? Werden alle externen Stakeholder (Kunde, Wettbewerb, SAP) beobachtet? Wer? Wie? Von wem?
- Sind alle erforderlichen **IT Service Management Prozesse** (Financial Management, Requirement Management, Change Management, Test Management, Release Management, Portfolio/Catalogue Management, …) definiert? Inkl. Verantwortlichkeiten, Zuständigkeiten, Schnittstellen, In- und Output?
- Welchen **Produktisierungsgrad** (Spezialisierung, Professionalisierung, Standardisierung, Modularisierung, Produktisierung) haben wir aktuell? Wo wollen wir hin? Wie kommen wir da hin?
- Welchen **Reifegrad** (Initial, Reproduzierbar, Synchronisiert, Definiert, Gesteuert, Optimiert, Wertschöpfend, Eigenständig) haben wir in der Entwicklung und den einzelnen Beratungsthemen? Wo wollen wir hin [um Add-ons bauen, supporten und weiterentwickeln zu können]? Wie kommen wir da hin?

Nachdem die Diskussion der Rahmenbedingungen abgeschlossen ist und ein genereller Konsens existiert, ist das weitere Vorgehen zu klären bzw. die Add-on-/Produktentwicklung konkret anzugehen:

- Schritt 1 – Schaffung der Grundlagen
  - Welcher Reifegrad wird benötigt und was ist konkret zu tun um ihn entsprechend zu erhöhen?

  - Welcher Produktisierungsgrad wird benötigt und was ist konkret zu tun um ihn zu erhöhen?
- Schritt 2 – Aufbau des Framerworks
  - Haben wir das Management-Commitment für die Add-on Entwicklung?
    Gibt es einen Business Plan? Werden durch Geschäftsführung Budget und Ressourcen bereitgestellt?
    Gibt es konkrete zeitliche und inhaltliche Ziele?
    Sind Verantwortlichkeiten + Zuständigkeiten definiert?
  - Welche Rahmenbedingungen sind im ersten Schritt konkret zu schaffen?
  - Was für eine Lösungslandschaft muss aufgebaut werden?
    Welches Template / Referenzmodell muss definiert und implementiert werden?
- Schritt 3 – Entwicklung und Pflege konkreter Add-ons
  - Welche Add-ons sind konkret zu pflegen (Ist) und zu entwickeln (Soll)?
    Wer treibt? Wer baut? Wer entwickelt? Wer verkauft?
- Schritt 4 – Entwicklung und pflege konkreter Produkte
  - …

…

Für eine tiefgehende Beschäftigung mit diesem Themengebiet sei auf andere Publikationen verwiesen.

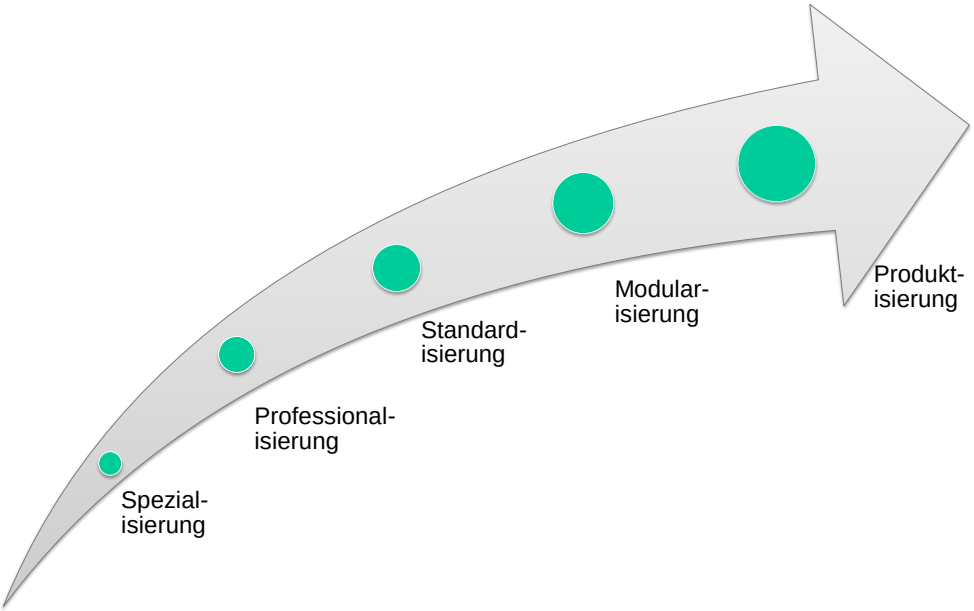
6.6 Spezialisierung, Professionalisierung und Standardisierung

Der Ausbau der Serviceorientierung geht zwangsläufig mit dem Ausbau der Produktisierung einher. Je höher der Produktisierungsgrad, desto höher ist auch der Grad der Serviceorientierung (und umgekehrt), desto höher sind Gewinnmarge und Kundenzufriedenheit, desto höher ist der Unternehmenserfolg...und das Unternehmensrisiko.

Eine echte Serviceorientierung und Produktisierung ist daher erst möglich, wenn Anwendungsberatung und -betreuung ausreichend spezialisiert, professionalisiert und standardisiert sind bzw. agieren. Welche Bedeutung diese Voraussetzung hat, wie eine entsprechende Selbstbewertung erfolgen könnte und welche Strategien, Ziele und Maßnahmen zu ergreifen sind, ist ein eigenständiges komplexes hochinteressantes Thema und würde ein eigenes Buch füllen. Die folgende Darstellung kann als Startpunkt einer intensiveren Beschäftigung mit dieser Thematik dienen:

Stufe	Beschreibung
Produktisierung	In der Informationstechnologie wird unter einem Produkt meist eine Komplettlösung bestehend aus Software, Hardware, Beratungs- und Betreuungsleistungen verstanden. Diese Komplettlösung ist multiplizierbar und bei mehreren Kunden einsetzbar. Die Kosten für Aufbau, Weiterentwicklung und Betrieb des Produktes entstehen nur einmal bzw. sind in der Folge deutlich geringer, was eine signifikant höhere Gewinnmarge für den Anbieter zur Folge hat. Gleichzeitig profitiert der Kunde von einer höheren Flexibilität und Reife der Lösung bzw. einer schnelleren Einführung und einem kostengünstigeren Betrieb.
Modularisierung	Ein Produkt besitzt eine mehrschichtige Architektur und besteht aus vielen kleinen Bausteinen. Diese Bausteine sind nach und nach entstanden, erst im Nachhinein zu einer Gesamtlösung zusammengesetzt wurden und werden im Laufe des Produktlebens erweitert, zurückgebaut, ersetzt oder bleiben nahezu unverändert. Voraussetzung für die Produktisierung der Anwendungsberatung und -betreuung ist damit die Modularisierung und damit einhergehend der Aufbau und der Betrieb lebensfähiger + kombinierbarer Produktbausteine bzw. -services.
Standardisierung	Die einzelnen Module / Bausteine sind eigenständig lebens- und vermarktungsfähig, ergeben jedoch nur innerhalb eines Kontextes (z.B. eine konkrete Funktionalität der Standardlösung oder ein konkreter Geschäftsprozess, die/der mit Hilfe des Bausteins effizienter oder effektiver unterstützt werden kann) einen Sinn. Für diesen Kontext besitzen Anwendungsberatung und -betreuung standardisierte Angebotsvorlagen, Konzepte, Dokumentationen und Präsentationen. Da eine Vermarktung von Bausteinen nur mit Kontext möglich ist, ist eine notwendige Voraussetzung für die Modularisierung die Standardisierung der Anwendungsberatung- und -betreuung.

Professionalisierung	Welche Zusatzentwicklung wird häufig benötigt? Gibt es Alternativlösungen im Standard oder von Wettbewerbern? Warum gibt es keine Alternativlösungen bzw. wieso ist die eigene Zusatzentwicklung besser? Handelt es sich um eine Zwischenlösung oder um eine dauerhafte Nische? ... Diese und zahlreiche weitere Fragen sind zu Beginn der Erstellung eines Produktbausteines zu klären. Voraussetzung für die Beantwortung dieser Fragen ist jedoch, dass der Anwendungsberater bzw. -betreuer als Experte auf seinem Gebiet tätig ist und die Geschäftsprozesse, die Standardlösung, die Konkurrenzlösungen und die Zukunft kennt.
Spezialisierung	Die IT-Welt ist sehr komplex. Entweder man ist Generalist oder Spezialist. Beides wird je nach Aufgabe benötigt. Um professionell Beratungs- bzw. Betreuungstätigkeiten durchführen zu können, ist zweifelsohne eine Spezialisierung auf das entsprechende Themengebiet erforderlich.



6.7 Steuerung und Verwaltung der Anwendungsentwicklung

Die Anwendungsentwicklung bedarf einer Verwaltung und Steuerung in der Linie (= dauerhafte Heimat) und im Einsatz (= temporäre Aufgabe). In diesem Handbuch haben wir das Thema Custom Code Lifecycle Management als Liniensteuerungs- und

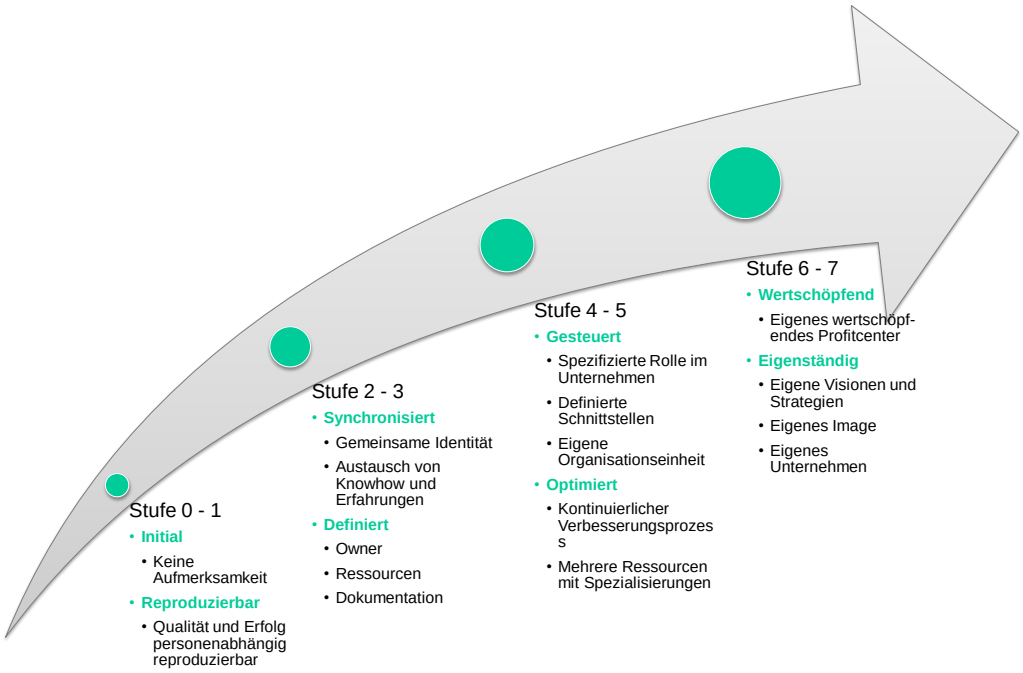
Linienverwaltungsframework kennen gelernt. Wie bereits in Abschnitt 6.1 erwähnt, ist diese Betrachtung vertiefungsfähig. Hinzu kommt, dass es ergänzende Steuerungs- und Verwaltungsmodelle gibt, die in diesem Handbuch nicht behandelt wurden, aber mindestens genauso wichtig sind. An dieser Stellen seien das Projektmanagement und das System/Software Engineering als Einsatzsteuerungs- und Einsatzverwaltungsframework genannt. Diese Modelle besitzen Überschneidungen / Präzisierungen und gegenseitige Ergänzungen mit dem IT Service Management / Custom Code Lifecycle Management, da ein Projektservice in gewisser Weise als ein spezieller IT-Service betrachtet werden kann. Das ITSM / CCLM ist ohne diese Präzisierungen / Ergänzungen nicht lebensfähig.

6.8 Standort- & Zielbestimmung / Ganzheitliche Betrachtung

Jeder Service, jeder Prozess und jede Aktivität erfahren im Laufe ihres Lebens einen Reifungsprozess. Es existieren dabei direkte und indirekte Korrelationen zwischen dem Nutzen für das Unternehmen und dem Reifegrad. Aus diesem Grund ist es wichtig, realistisch einzuschätzen, welcher Reifegrad aktuell erreicht ist und welcher maximal erreichbar scheint.

Zu Beginn ist der Reifegrad sehr gering, steigt aber im Laufe der Zeit kontinuierlich an. Es ist durchaus möglich einige Stufen zu überspringen oder lange Zeit auf einer Stufe zu verweilen. Sogar Rückschritte kann es geben. Und in fast allen Fällen wird der höchste Reifegrad aufgrund äußerer Rahmenbedingungen niemals erreicht werden.

Es ist nicht schlimm, dass das Maximum nie erreicht wird. Wichtig ist jedoch die Erreichung des Minimums, denn eine Wertlieferung zum Kunden ist erst ab einem ausreichend hohen Reifegrad (>= 4) servicefähig. Ein Service ist nur dauerhaft lebensfähig, wenn zudem die einzelnen IT Service Management Prozesse, das IT Service Management als Ganzes, die konsumierenden und die konsumierten Services ebenfalls einen ausreichend hohen Reifegrad (>= 4) besitzen.



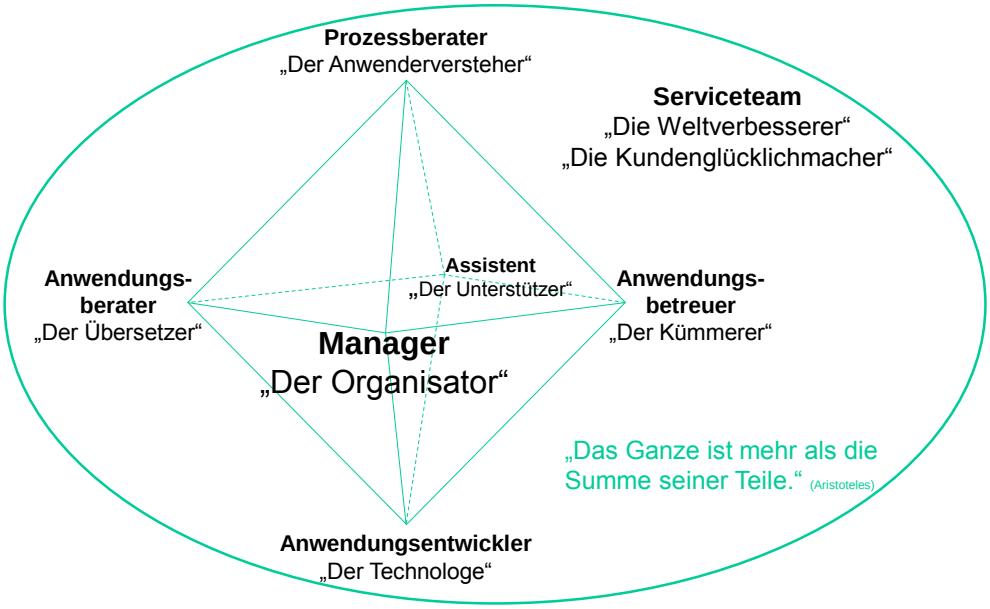
Im Folgenden werden die Reifegrade als Basis für eine Ist-Analyse "Wo stehe ich gerade?" und Soll-Analyse "Wo will ich hin?" kurz vorgestellt. In diesem Zusammenhang sei erwähnt, dass eine Zuordnung zu einem Reifegrad nicht immer eindeutige möglich ist, die folgenden Beschreibungen liefern somit lediglich Anhaltspunkte. Die Beschreibungen beziehen sich auf einen konkreten "Prozess". Die Aussagen gelten aber auch für "Services" und einzelne "Aktivitäten".

Stufe	Beschreibung
Stufe 0: Initial	Ein Prozess auf dieser Stufe wird grundsätzlich zwar als Prozess erkannt, allerdings gibt es keinerlei Aktivitäten für seine kontrollierte Abarbeitung. Darüber hinaus werden für den Prozess keine Ressourcen zur Verfügung gestellt, und es wird ihm in der Organisation keine Bedeutung oder Aufmerksamkeit geschenkt.
Stufe 1: Reproduzierbar	Es ist in dem Unternehmen ein Bewusstsein dafür vorhanden, dass es einen solchen Prozess gibt. Aufmerksamkeit, zugemessene Bedeutung und zugewiesene Ressourcen sind aber weiterhin sehr gering. Grundlegende Aktivitäten des Prozesses können beschrieben sein, müssen aber nicht. Der Prozess liefert allerdings schon in diesem Stadium reproduzierbare Ergebnisse. Qualität und Erfolg sind jedoch stark personenabhängig. Das Thema wird im Unternehmen an verschiedenen Stellen unabhängig voneinander nebenbei angegangen. Die Ressourcen bilden keine eigene Organisationseinheit und bearbeiten nebenbei auch andere Aufgaben.
Stufe 2: Synchronisiert	Es entwickelt sich eine eigene Identität. Die Ressourcen betreiben zur Effizienzsteigerung und zur Nutzung von Synergien einen Informations- und Wissensaustausch in Form von Competence Teams. Im Tagesgeschäft agieren die Ressourcen jedoch weitestgehend unabhängig voneinander.
Stufe 3: Definiert	Der Prozess ist dokumentiert, seine Rolle im Unternehmen ist aber nicht spezifiziert. Ebenso gibt es keine formale Vereinbarung zu dem Prozess. Allerdings gibt es auf dieser Stufe bereits einen Process Owner, also jemanden, der für die Qualität des Prozesses verantwortlich ist, sowie zugewiesene Ressourcen. Darüber hinaus ist der Prozess primär auf Effektivität und Effizienz und weniger auf Qualität ausgerichtet. Die Steuerung und Kontrolle erfolgt in sogenannten Competence Centern als virtuelle Heimat. Die Ressourcen agieren nach innen und außen als Einheit.

Stufe 4: Gesteuert	Der Prozess ist in der Organisation akzeptiert und vollständig dokumentiert. Seine Ziele basieren auf den Geschäfts-anforderungen und sind auf die Services der Organisation ausgerichtet. Der Prozess läuft zunehmend proaktiv ab und verfügt über fest definierte Schnittstellen zu anderen Prozessen. Die Ressourcen sind in Vollzeit permanenten zugewiesen, bilden eine eigenständige Gruppe bzw. ein Team und besitzen somit in der Organisationsstruktur eine reale Heimat.
Stufe 5: Optimiert	Prozesse, die auf diesem Reifegrad betrieben werden, verfügen über einen kontinuierlichen Verbesserungsprozess mit dem Ziel, den Prozess ständig neu auf die Unternehmensziele auszurichten. Darüber hinaus beschäftigen sich bereits mehrere Ressourcen mit dem Prozess, so dass zunehmend eine Spezialisierung möglich ist. Dem personellen Wachstum und der Spezialisierung wird durch Zerlegung in mehrere Gruppen / Teams und Gründung eines eigenen Bereiches in der Organisationsstruktur Rechnung getragen.
Stufe 6: Wertschöpfend	Ein weiteres personelles Wachstum und eine weitere Spezialisierung ermöglicht die Erschließung von Rand- und Nischenthemen. Darüber hinaus haben einige Gruppen / Teams aufgrund ihrer hohen Spezialisierung keine bzw. kaum noch Berührungspunkte zueinander. Es erfolgt daher die Zerlegung in mehrere Bereiche und die Gründung einer eigenen Geschäftseinheit. Diese Geschäftseinheit bildet ein eigenes Profitcenter und ist ein essentieller (wertschöpfender) Bestandteil des Unternehmens.
Stufe 7: Eigenständig	Die Geschäftseinheit entwickelt eigene Visionen und Strategien. Sie besitzt ein eigenes externes und internes Image und benötigt keine Subventionen durch andere Geschäftseinheiten. Teile der unterstützenden Organisationseinheiten (Buchhaltung, Vertrieb, etc.) sind ausschließlich für diese Geschäftseinheit tätig. Es erfolgt die Ausgründung als rechtlich eigenständiges (Tochter-)Unternehmen.

Ganz wichtig: Wie in Abschnitt 3.3 angedeutet, zählt die Anwendungsentwicklung zu den am stärksten vernetzten Abteilungen eines Unternehmens. Die Themen Serviceorientierung + Produktisierung, sowie die damit im direkten und indirekten Zusammenhang stehenden Themen, können daher nur unternehmensweit (alle zum Netzwerk der Anwendungsentwicklung gehörenden Abteilungen) eingeführt / optimiert / ausgebaut werden. Es macht zum Beispiel keinen Sinn, dass die Anwendungsentwicklung Produkte bauen möchte, die Anwendungsberatung jedoch keine Produktideen beisteuern kann und nicht in der Lage ist, die Produkte zu vermarkten. Oder die Anwendungsbetreuung diese Produkte nicht pflegen und

weiterentwickeln würde. Oder wenn die Anwendungsentwicklung "Operational Level Agreements" (unternehmensinterne Servicevereinbarungen) mit den anderen Abteilungen vereinbaren möchte, diese Abteilungen hieran aber gar kein Interesse haben bzw. ihre OLAs gegenüber der Anwendungsentwicklung nicht einhalten würden. Eine ganzheitliche Betrachtung ist daher unabdingbar:



Rolle	Beschreibung
Prozessberater	Das Analysieren, Modellieren, Visualisieren und Optimieren von Geschäftsprozessen zählt zu den Hauptaufgaben des Prozessberaters. Der Prozessberater besitzt Kenntnis über das jeweilige Fachthema und die Branche, er besitzt keine Kenntnisse über die Technologie. Er spricht die Sprache des Anwenders und ist ein "Anwenderversteher". *Die Prozessberatung berät den Anwender und erkennt hierbei die Notwendigkeit einer IT-Unterstützung.*
Anwendungsberater	Fachliche Konzeption (inkl. Businessanalyse), Technische Konzeption und Realisierung im Standard (mittels Customizing und Stammdatenpflege) sind typische Aktivitäten des Anwendungsberaters. Der Anwendungsberater besitzt Kenntnis über das jeweilige Fachthema und die Branche. Er besitzt darüber hinaus Kenntnisse über die einzusetzende Technologie. Er ist das konzeptionelle Bindeglied zwischen Prozessberater, Anwender und IT. *Die Anwendungsberatung berät den Anwender zu einer spezifischen Anwendungslösung und generiert hierbei Aufgaben für Anwendungsentwicklung.*
Anwendungs-entwickler	Der Anwendungsentwickler ist für die Technische Konzeption und Realisierung von Anpassungen und Erweiterungen zuständig. Er besitzt keine Kenntnis über Fachthema und Branche, dafür jedoch umfangreiche Kenntnisse über die Technologie. Er spricht die Sprache der IT und ist ein innovativer Technologe. *Anwendungsberatung und Anwendungsentwicklung realisieren gemeinsam eine nützliche Anwendungslösung mit langfristigem Betreuungsbedarf.*

Anwendungsbetreuer	Anwendungsbetreuer betreuen die IT-Lösung (Wartung, Ausführen von Jobs, Fehlersuche und -beseitigung, Kleine Anpassungen und Erweiterungen). Sie halten den Kundenkontakt und Generieren neue Beratungsprojekte. Sie besitzen Kenntnis über Fachthema, Branche und Technologie. Sie bilden das analytische Bindeglied zwischen Anwender und IT. *Die Anwendungsbetreuung betreibt die Anwendungslösung, betreut die Anwender und generiert hierdurch neue Projekte für die Prozessberatung.*
Manager	Prozessberater, Anwendungsberater, Anwendungsbetreuer und Anwendungsentwickler sind zuständig, der Manager ist dagegen verantwortlich. Die Organisation der Arbeit ist die aktive Aufgabe des Managers. Gleichzeitig nimmt er die passive Aufgabe der Eskalationsebene war. Manager besitzen keine Kenntnisse über Fachthema, Branche und Technologie. Stattdessen besitzen sie umfangreiche Kenntnisse über Projektmanagement, Mitarbeiterführung, Konfliktmanagement, ... In einem Unternehmen gibt es viele verschiedene Manager, deren konkrete Aufgaben bzw. Verantwortungsbereiche variieren: Projektmanager, Prozessmanager, Service Manager, Abteilungsleiter, Geschäftsführer. *Das Management steuert das Serviceteam.*
Assistenz	Die Assistenz ist zuständig für das Ausführen unterstützender Tätigkeiten. Sie hat nur indirekt Kontakt zur Projekt- bzw. Betriebsarbeit. Assistenten besitzen keine Kenntnisse über Fachthema, Branche, Technologie. Sie dienen der Unterstützung / Entlastung aller anderen Mitarbeiter. Zur Assistenz zählen unter anderem Projektassistenz und Sekretariat sowie Angestellte in der technischen und kaufmännischen Verwaltung. *Die Assistenz verwaltet und unterstützt das Serviceteam.*

Jede Rolle wird durch Ressourcen eines für diese Aufgabe spezialisierten Services besetzt. Jeder Service erbringt bei einer hinreichend hohen Reife eine hervorragende Dienstleistung, aber erst durch Vernetzung entsteht in der Summe ein Mehrwert für den Kunden: "Das Ganze ist mehr als die Summe seiner Teile."