



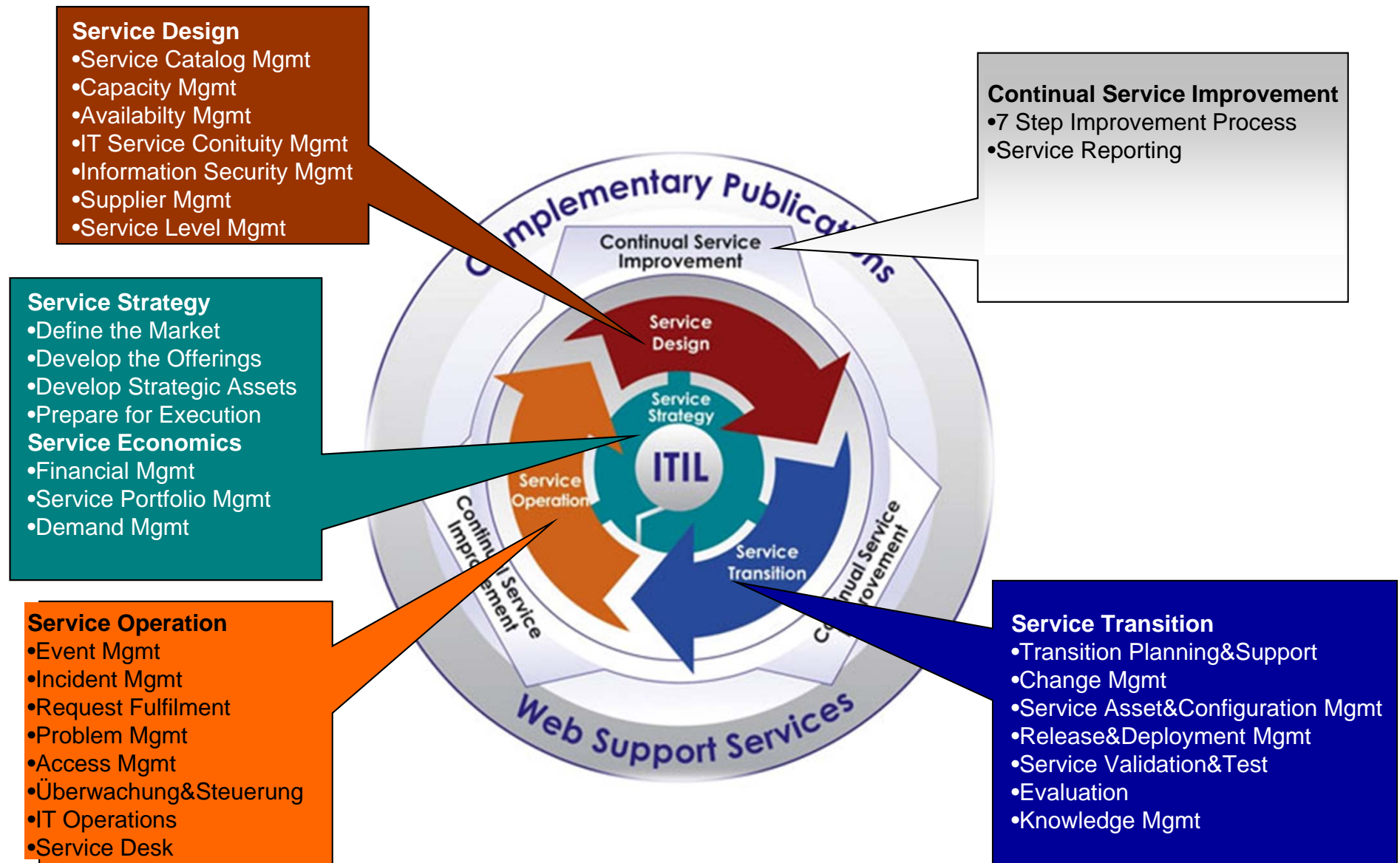
# IT Service Management

## Service Transition



Professor Dr. Mathias Philipp

### 3. Service Transition Grundbegriffe



## 3. Service Transition Grundbegriffe

### Service Transition

- Stellt Richtlinien für die Entwicklung, Verbesserung und qualifizierten Übergabe von **neuen oder geänderten Services** für den operativen Betrieb zur Verfügung.
- Für den operativen Betrieb notwendigen Prozesse, Systeme und Funktionen gemäß den Anforderungen aus dem Service Design.

### Ziele Service Transition

- Geordnete Überführung neuer oder geänderter Services in den Produktivbetrieb
- Definition und Bereitstellung der Test-, Release- und Kommunikationspläne
- Durchführung
  - ◆ Risiko- und Qualitätsmanagement
  - ◆ Tests
  - ◆ Betriebsübergabe
  - ◆ Early Life Support (ELS)
  - ◆ Geregelter Beenden der ELS-Phase über entsprechende Kriterien
- Unterstützung des organisatorischen Change beim Kunden

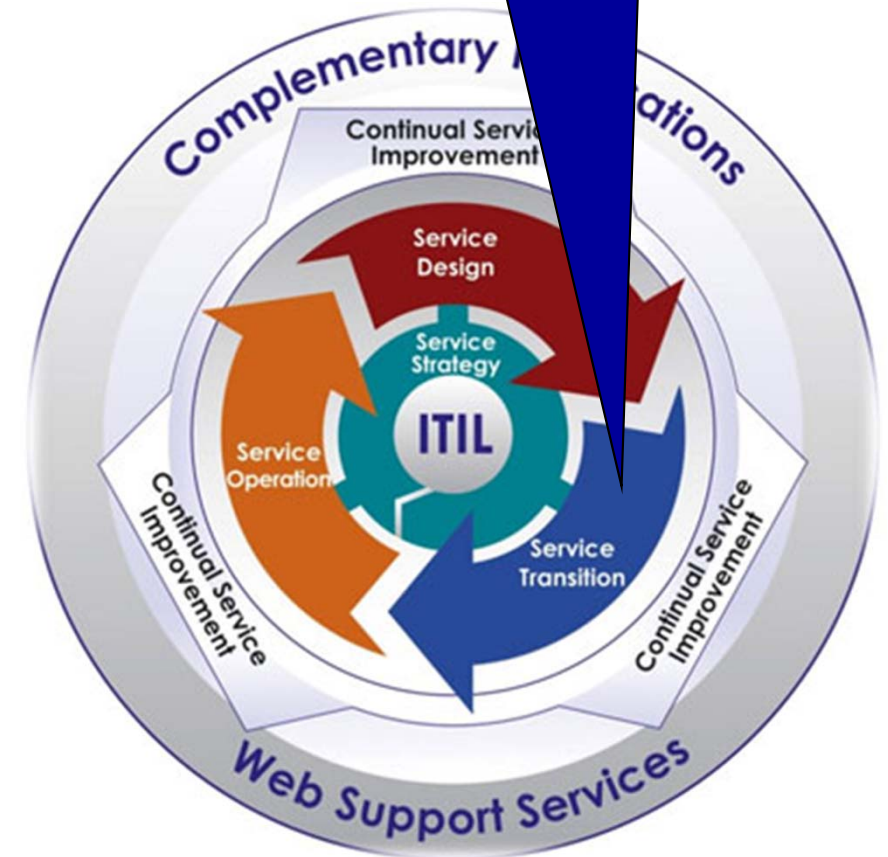
### 3. Service Transition Prozesse



- 3.1 Transition Planning&Support
- 3.2 Change Management
- 3.3 Service Asset&Configuration Management
- 3.4 Release&Deployment Management
- 3.5 Service Validation&Test
- 3.6 Evaluation
- 3.7 Knowledge Management

#### Service Transition

- Transition Planning&Support
- Change Mgmt
- Service Asset&Configuration Mgmt
- Release&Deployment Mgmt
- Service Validation&Test
- Evaluation
- Knowledge Mgmt



## 3. Service Transition

### 3.1 Transition Planning&Support

#### Zielsetzung

- Planung und Koordination aller Service-Transition-Prozesse und die hierfür benötigten Ressourcen.
  
  - Service-Transition-Prozesse:
    - ◆ Change Management
    - ◆ Service Asset and Configuration Management
    - ◆ Knowledge Management
    - ◆ Transition Planning and Support
    - ◆ Release and Deployment Management
    - ◆ Service Validation and Testing
    - ◆ Evaluation
- } Prozesse unterstützen den gesamten Lebenszyklus

} Prozesse unterstützen primär die Transition Phase

## 3. Service Transition

### 3.1 Transition Planning&Support

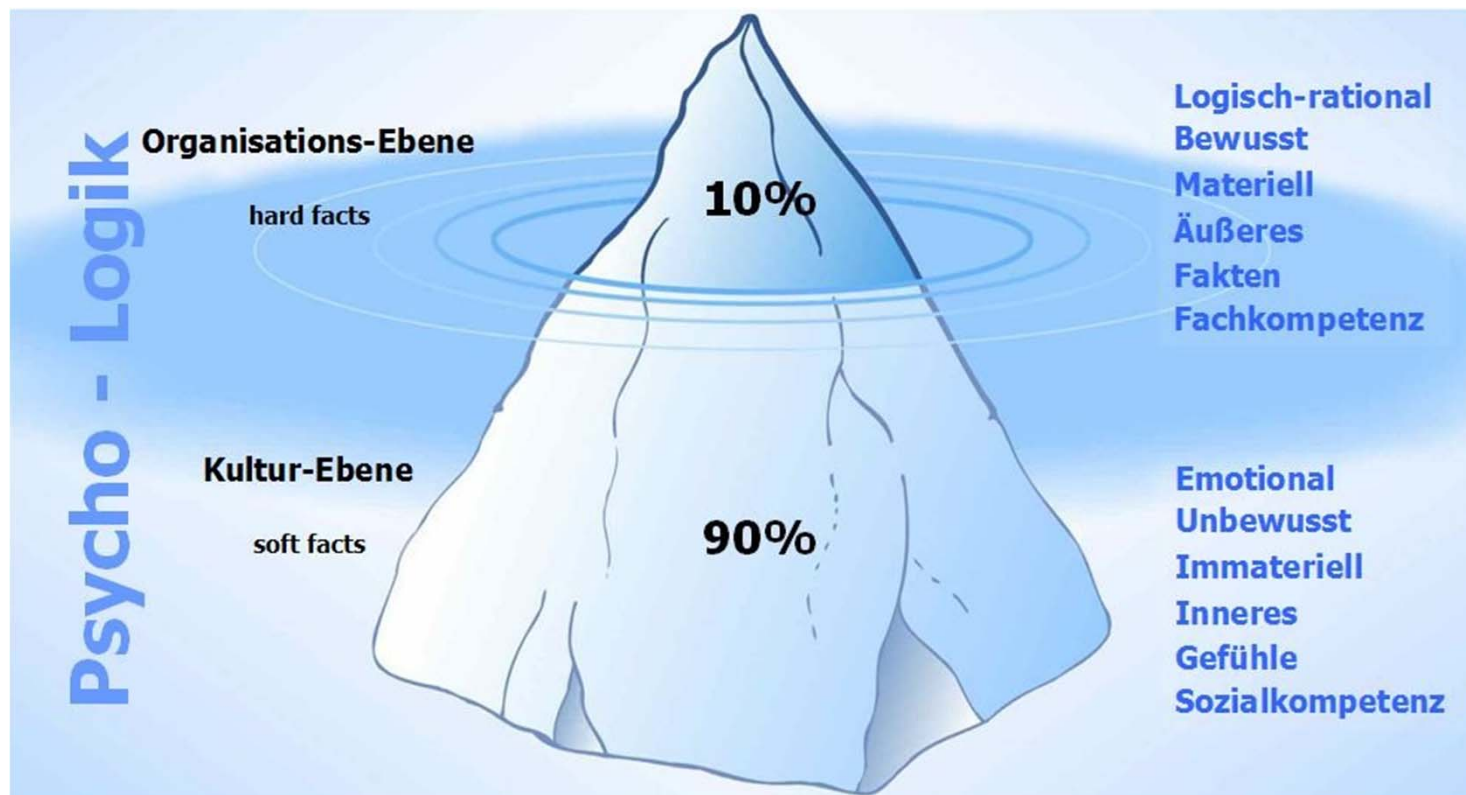
## Grundbegriffe

### Service Design Package (SDP)

- Enthält alle Anforderungen eines IT Services über alle Lebenszyklusphasen
- Wurde in der Service Design Phase entwickelt
- Für die Transition Phase enthält das SDP z.B. folgenden Informationen:
  - ◆ Service-Spezifikationen, Service-Pakete, Service-Abnahmekriterien
  - ◆ Architektur-Design (einschließlich der Service-Komponenten)
  - ◆ Release- und Deployment-Pläne

## 3. Service Transition

### 3.2 Change Management



Das Eisberg-Prinzip

## 3. Service Transition

### 3.2 Change Management

#### Zielsetzung

- Sicherstellen, dass **Änderungen** am IT-Betrieb unter dem Aspekt der Risikominimierung effizient und mit möglichst geringen negativen Auswirkungen auf die Qualität der IT-Serviceerbringung umgesetzt werden.

#### Aufgaben

- Stellt standardisierte Methoden und Verfahren zur Bearbeitung von Änderungen zur Verfügung,
- so dass Änderungen sicher und kontrolliert durchgeführt werden können und
- sich so wenig wie möglich auf die Qualität der IT-Services auswirken.
- Planung und Kontrolle alle Änderungen an der IT-Infrastruktur.



## 3. Service Transition

### 3.2 Change Management

#### Grundbegriffe

##### ■ Änderungsantrag (Request for Change RFC)

Ein RFC ist Auslöser des Change Management Prozesses und kann von allen beteiligten Prozessen gestellt werden.

- ◆ Beschreibung Ursache (Known Error) oder Anlass der Änderung
- ◆ Lösungsalternativen

##### ■ Change

- ◆ Hinzufügen, Modifizieren oder Entfernen eines Elements, das Auswirkungen auf die IT Services haben könnte. Der Umfang eines Change sollte sämtliche IT Services, Configuration Items, Prozesse, Dokumentationen etc. einschließen.
- ◆ Ein autorisierter Änderungsantrag (RFC) wird als Veränderung (**change**) bearbeitet und nach einer Überprüfung (Post Implementation Review) geschlossen.

## 3. Service Transition

### 3.2 Change Management

#### Changearten

- Standard Change
  - ◆ **vorautorisierter** standardisierbarer Change
  - ◆ Häufiges Auftreten
  - ◆ geringes Risiko, definierte Auswirkungen
  - ◆ Vereinfachte standardisierte Changebearbeitung
  - ◆ Ggf. im First-Level-Support zur Erhöhung der First-Resolution-Rate
- Normal Change
  - ◆ Change mit gewisser Dinglichkeit und Komplexität
  - ◆ Übergeordnete Koordination erforderlich
  - ◆ Genehmigung über das Change Advisory Board (CAB)
- Emergency Change (ECAB)
  - ◆ Change mit höchster Dringlichkeit und kurzfristigem Änderungsbedarf (gravierende Störungen)
  - ◆ Changebearbeitung auf Basis klarer und kurzer Prozessstufen unter starker Managementbeobachtung

## 3. Service Transition

### 3.2 Change Management

#### Change Advisory Board (CAB)

- Das CAB berät den Change Manager bei seiner Entscheidungsfindung. Die Endverantwortung hat der Change Manager.
- Das CAB besteht aus permanenten und für die jeweilige Veränderung (Change) vorgeschlagen Mitgliedern. Es tagt regelmäßig. Typische Zusammensetzung
  - ◆ Change Manager (Vorsitzender)
  - ◆ Vertreter aus dem Incident, Problem- und Release-Management
  - ◆ Vertreter der Anwendungsentwicklung
  - ◆ IT-Spezialisten
  - ◆ Bereichsmanager und Manager der Finanzabteilung
  - ◆ Vertreter der Kundenumgebung
  - ◆ Vertreter der Dienstleister

#### Emergency Change Advisory Board (ECAB)

- wird in Krisensituationen oder bei dringenden Änderungen einberufen
- Teilgruppe des CAB (Rufbereitschaft)

## 3. Service Transition

### 3.2 Change Management

#### **Schedule of Change (SC)**

- Liste der (genehmigten) Changes
- Planungsinstrument des CAB
- Veröffentlichung des Zeitplans der genehmigten RFCs

#### **Rückfallplan (Back-out/Fallback)**

- Keine Genehmigung eines Change ohne Back-out Plan.
- tritt bei unvorhersehbaren Schwierigkeiten in Kraft und enthält detaillierte Anweisungen, um auf die letzte funktionierende Stufe (Baseline) zurückzukehren.

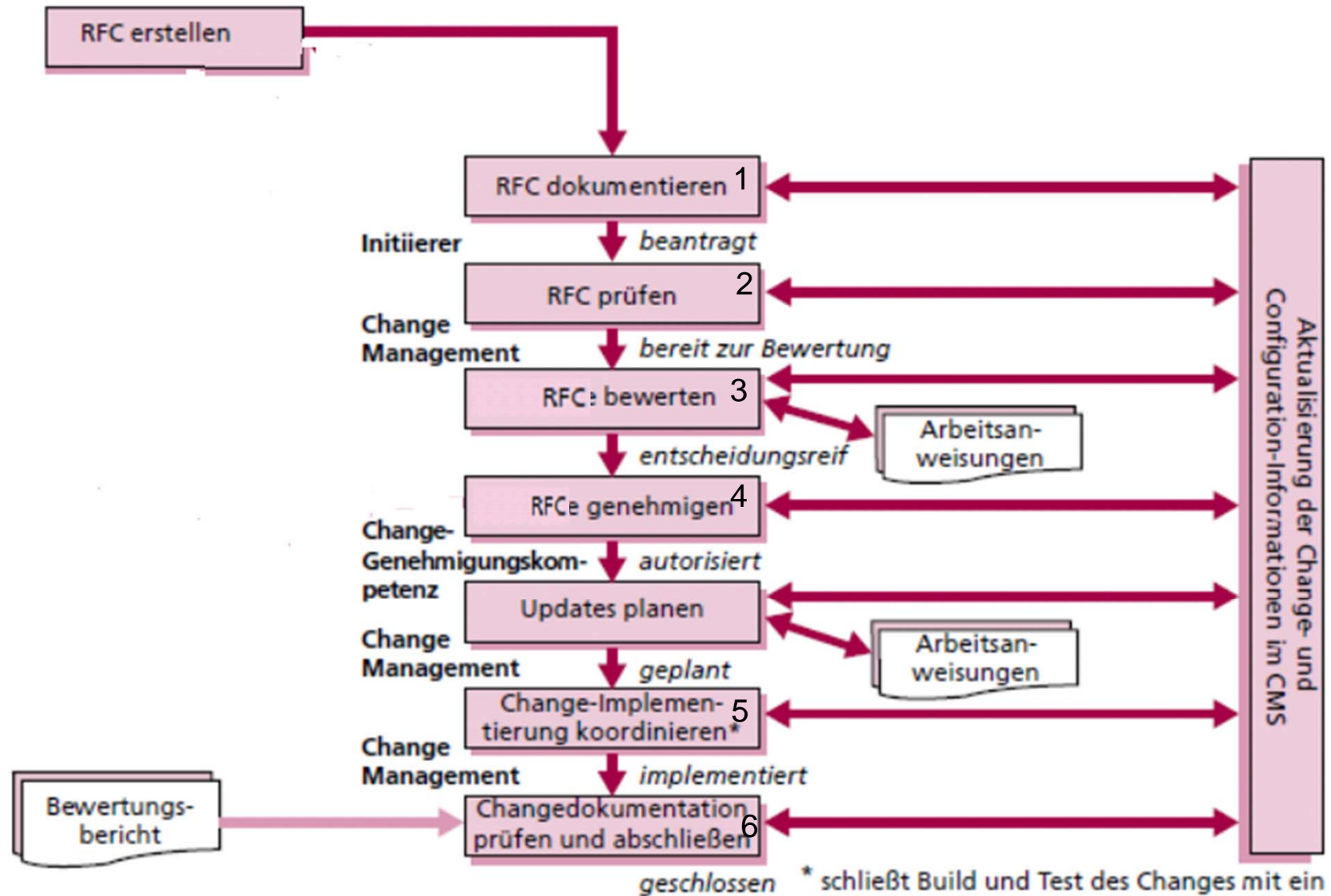
#### **Post Implementation Review (PIR)**

- Ein Review, der nach der Implementierung eines Change oder eines Projekts erfolgt. Ein PIR stellt fest, ob der Change oder das Projekt erfolgreich ist, und identifiziert Verbesserungsmöglichkeiten.

## 3. Service Transition

### 3.2 Change Management

#### Beispiel 1: Prozessfluss eines Change / Aktivitäten Change Management



## 3. Service Transition

### 3.2 Change Management

#### Beispiel 1: Prozessfluss eines Change / Aktivitäten Change Management

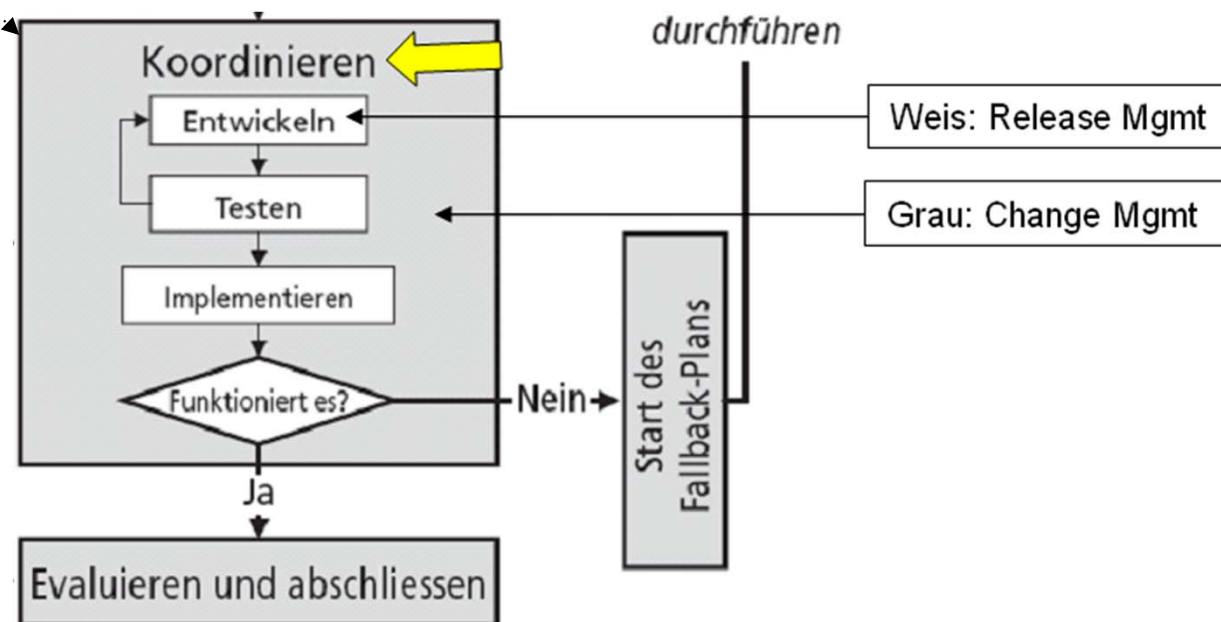
1. RfC dokumentieren:  
Alle eingereichten RfCs werden mit eindeutiger Kennung erfasst
2. RfC überprüfen:  
Prüfung auf Vollständigkeit, Konsistenz oder schon früher erfasst
3. RfC bewerten:  
Machbarkeit, Kosten- / Nutzen => danach Entscheidungsvorlage CAB
4. RfC genehmigen:  
Autorisierung des RfC => Change

## 3. Service Transition

### 3.2 Change Management

#### Beispiel 1: Prozessfluss eines Change / Aktivitäten Change Management

5. Koordination der Implementierung: Umsetzung und Test des Change und Weiterverteilung als Release

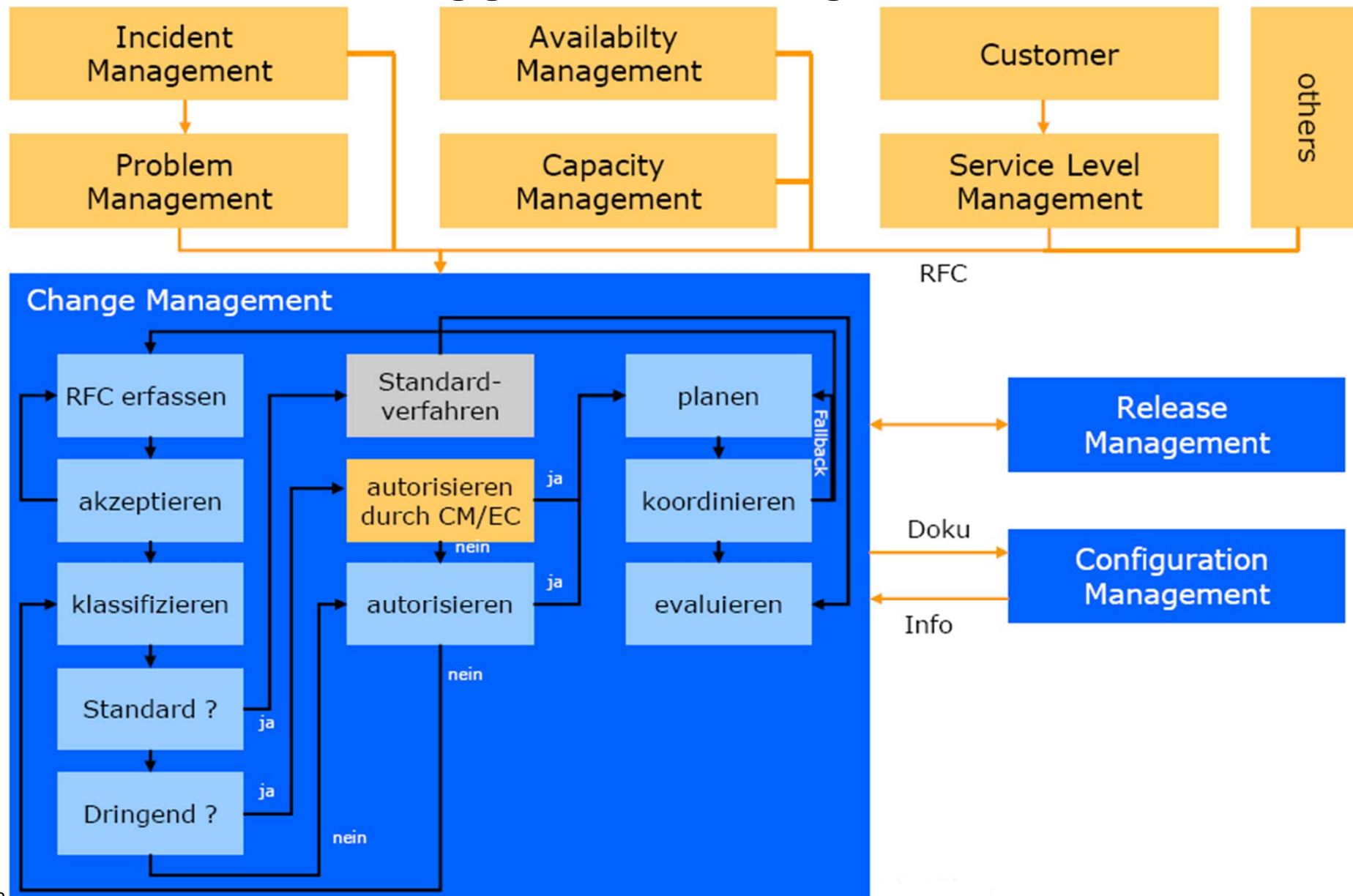


6. Bewerten und Beenden: Post Implementation Review hinsichtlich Erfolg des Change und Closing

## 3. Service Transition

### 3.2 Change Management

#### Beispiel 2: Prozessfluss in Abhängigkeit von der Changeart





**Welche der folgenden Informationen über eine bereits ausgeführte Änderung (Change) ist im Change Management Teil der Berichterstattung an die Leitung der Organisation?**

- A.** Anzahl der Störungen (Incidents) bezogen auf die durchgeführte Änderung (Changes)
- B.** Anzahl der gelösten Störungen (Incidents) aufgrund der durchgeführten Änderung
- C.** Falsch registrierte Configuration Items (CIs)
- D.** Aufbau und Zusammenstellung der Configuration Items (CIs)

**Welche der folgenden Informationen über eine bereits ausgeführte Änderung (Change) ist im Change Management Teil der Berichterstattung an die Leitung der Organisation?**

- A.** Anzahl der Störungen (Incidents) bezogen auf die durchgeführten Änderungen (Changes)
  - B.** Anzahl der gelösten Störungen (Incidents) aufgrund durchgeführter Änderungen (Changes)
  - C.** Falsch registrierte Configuration Items (CIs)
  - D.** Aufbau und Zusammenstellung der Configuration Items (CIs)
- 
- A.** Richtig. Dies ist ein Hinweis für die Wirksamkeit des Change Managements.  
[Hinweis: Die Anzahl der durch einen Change erzeugen (neuen) Störungen]
  - B.** Falsch. Die Anzahl der gelösten Störungen (Incidents) ist in der Berichterstattung des Problem Managements oder des Incident Managements enthalten.  
[Sinnvoll als Indikator für Workload und ausreichende Personalkapazitäten]
  - C.** Falsch. Das gehört zum Aufgabenbereich des Configuration Managements.
  - D.** Falsch. Das gehört zum Aufgabenbereich des Configuration Managements.



**Welcher Prozess liefert Input zur Einschätzung der Auswirkung (Impact) einer Änderung (Change) an das Change Management?**

- A.** Configuration Management
  - B.** Incident Management
  - C.** Problem Management
  - D.** Release Management
-

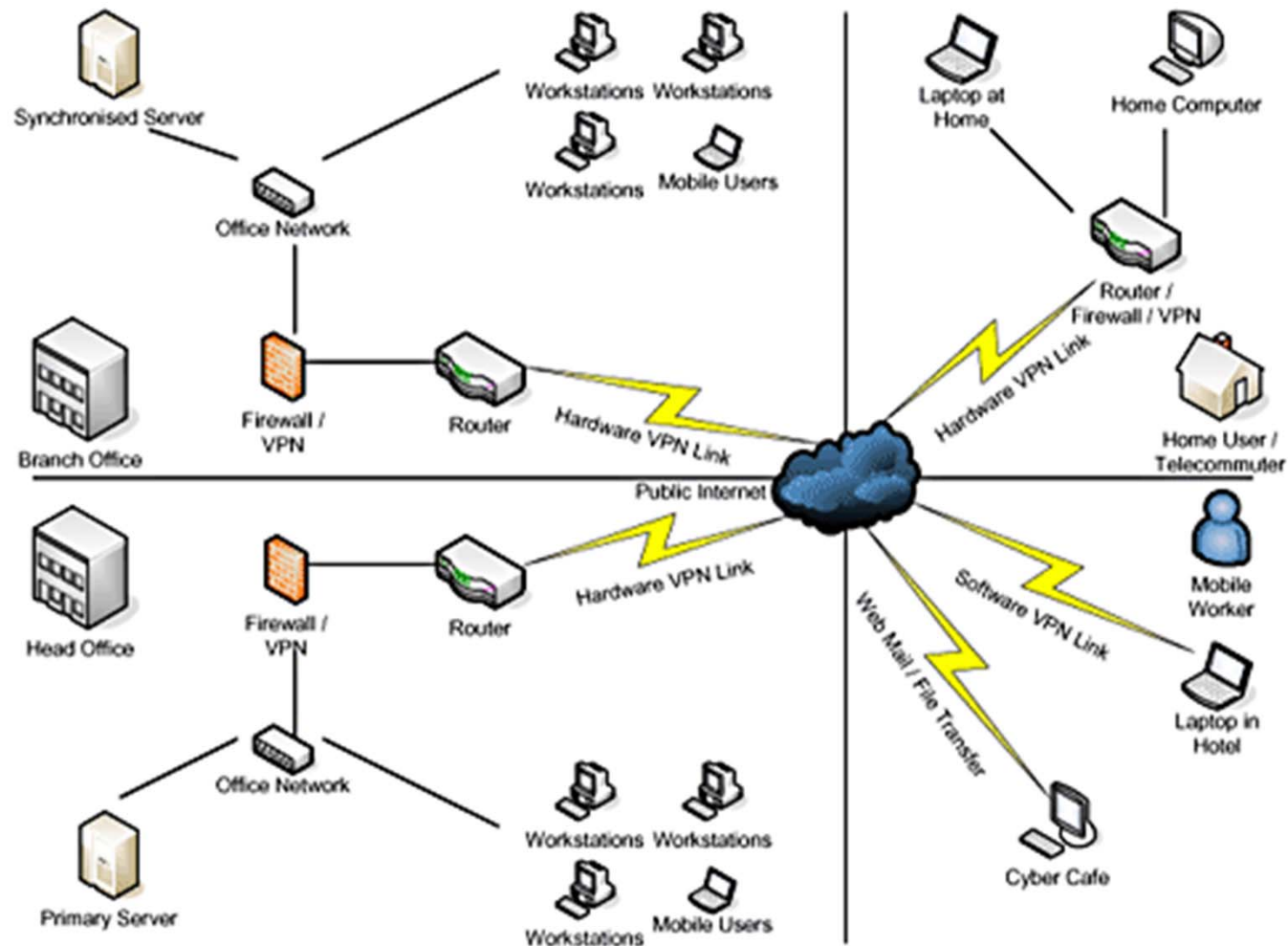
**Welcher Prozess liefert Input zur Einschätzung der Auswirkung (Impact) einer Änderung (Change) an das Change Management?**

- A. Configuration Management**
  - B. Incident Management**
  - C. Problem Management**
  - D. Release Management**
- 

- A. Richtig. Prozessinputs des Change Management sind: RFC, CMDB, Schedule Change. Die CMDB liefert Zusammenhänge/Beziehungen zwischen CIs. Dies dient der Einschätzung der Auswirkung eines Change auf andere CI und die mit den CIs verbundenen Services.
  - B. Falsch. Es gibt auch RFCs, die nicht über das Incident Management kommen.
  - C. Falsch. Es gibt aber auch RFCs, die nicht über das Problem Management kommen.
  - D. Falsch. Das Release Management liefert kein Input an das Change Management.
- Einstieg über Auswirkung – Priorisierung/Klassifizierung führt zu Lösung Incident Mgmt

## 3. Service Transition

### 3.3 Asset&Configuration Management



## 3. Service Transition

### 3.3 Asset&Configuration Management

#### Zielsetzung

- Bereitstellung eines logischen Modells der IT-Infrastruktur.
  - ◆ In diesem Modell werden die IT Services zu den unterschiedlichen, zur Lieferung dieser Services benötigten IT-Komponenten (CIs) in Beziehung gesetzt.
- Aufbau und Pflege des Configuration Management System (CMS) bzw. der CMDB

## 3. Service Transition

### 3.3 Asset&Configuration Management

#### Grundbegriffe

- **Configuration Item (CI)**  
Ein CI ist ein Asset, eine Service Komponente oder irgendein anderes Objekt, das vom Configuration Management verwaltet wird oder werden soll.  
Beispiel: Services, Hardware, Software, aktive/passive Netzwerkkomponenten, Dokumentationen, Lizenzen, Verfahren, SLAs, Kunden, Server, **Mitarbeiter**
- **Attribut**  
Information zur Beschreibung eines CI (z.B. Versionsnummer, Standort)
- **Beziehung**  
logische Verbindung oder Abhängigkeit zwischen zwei CIs  
z.B. „Stückliste“, „Verwendungsnachweis“
- **Configuration**  
ist alles, was beherrscht werden soll (z.B. Hardware, Service)
- **Configurations Struktur**  
zeigt die Beziehungen und Hierarchien (Struktur) der CIs einer Configuration
- **Baseline**  
ist eine Ausgangs-Konfiguration, die aus einer Gruppe von „eingefrorenen“ CIs besteht (z.B. Standardarbeitsplatz, Werkseinstellung, Image)

## 3. Service Transition

### 3.3 Asset&Configuration Management

#### Grundbegriffe

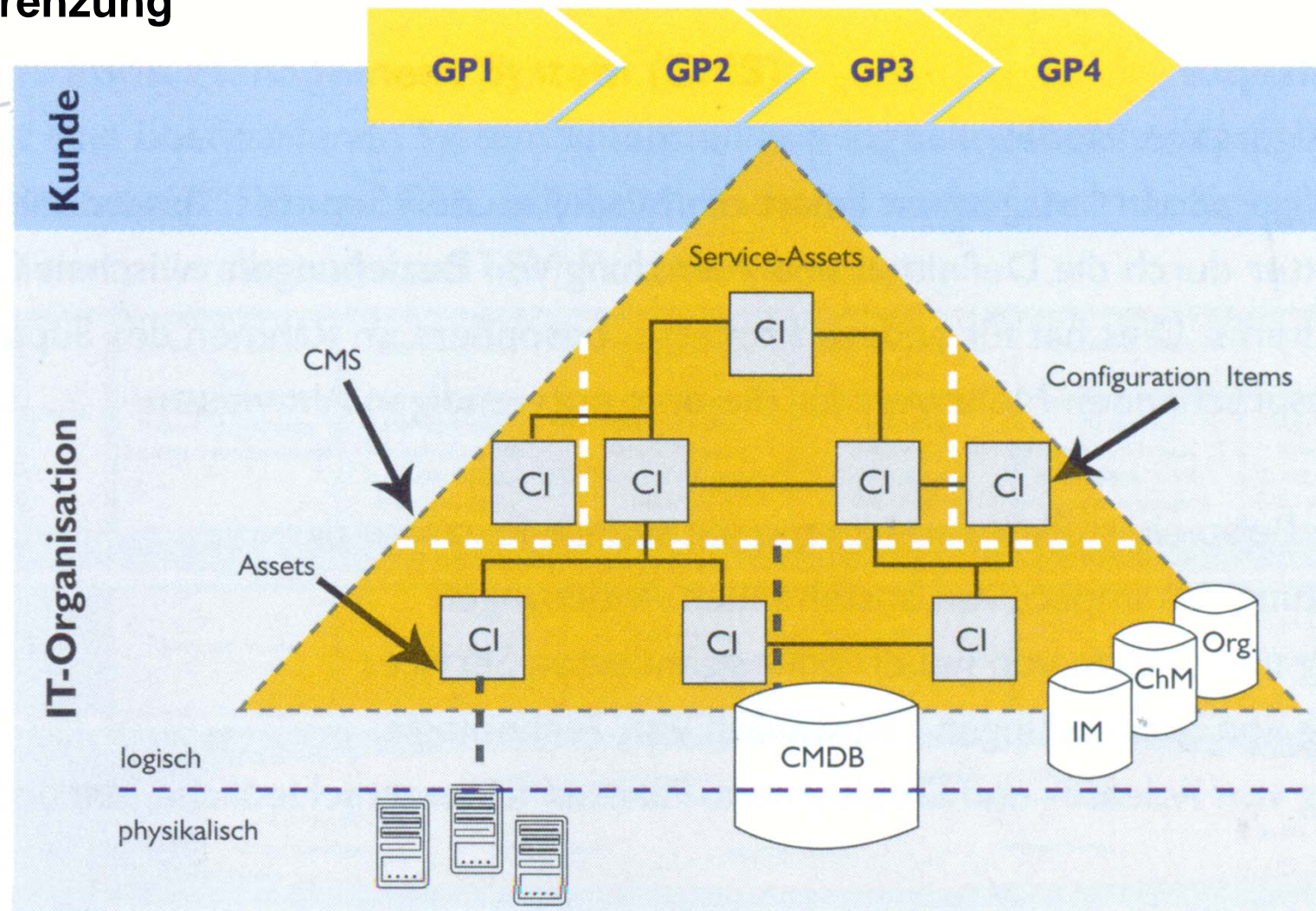
- **Configuration Management Database (CMDB)**  
logisches Modell der Infrastruktur zur Pflege der CIs und deren Beziehungen
  - ◆ Z.B. welche CIs benötigt ein Service? („Produktstückliste“)
  - ◆ Z.B. welche CIs/Services sind bei Ausfall eines CI betroffen-Auswirkungsanalyse? („Verwendungsnachweis“)
  
- **Configuration Management System (CMS)**  
CMS = ein oder mehrere CMDB



## 3. Service Transition

### 3.3 Asset&Configuration Management

#### Begriffsabgrenzung



Quelle: Learn IT!Lv3, S.105

## 3. Service Transition

### 3.3 Asset&Configuration Management

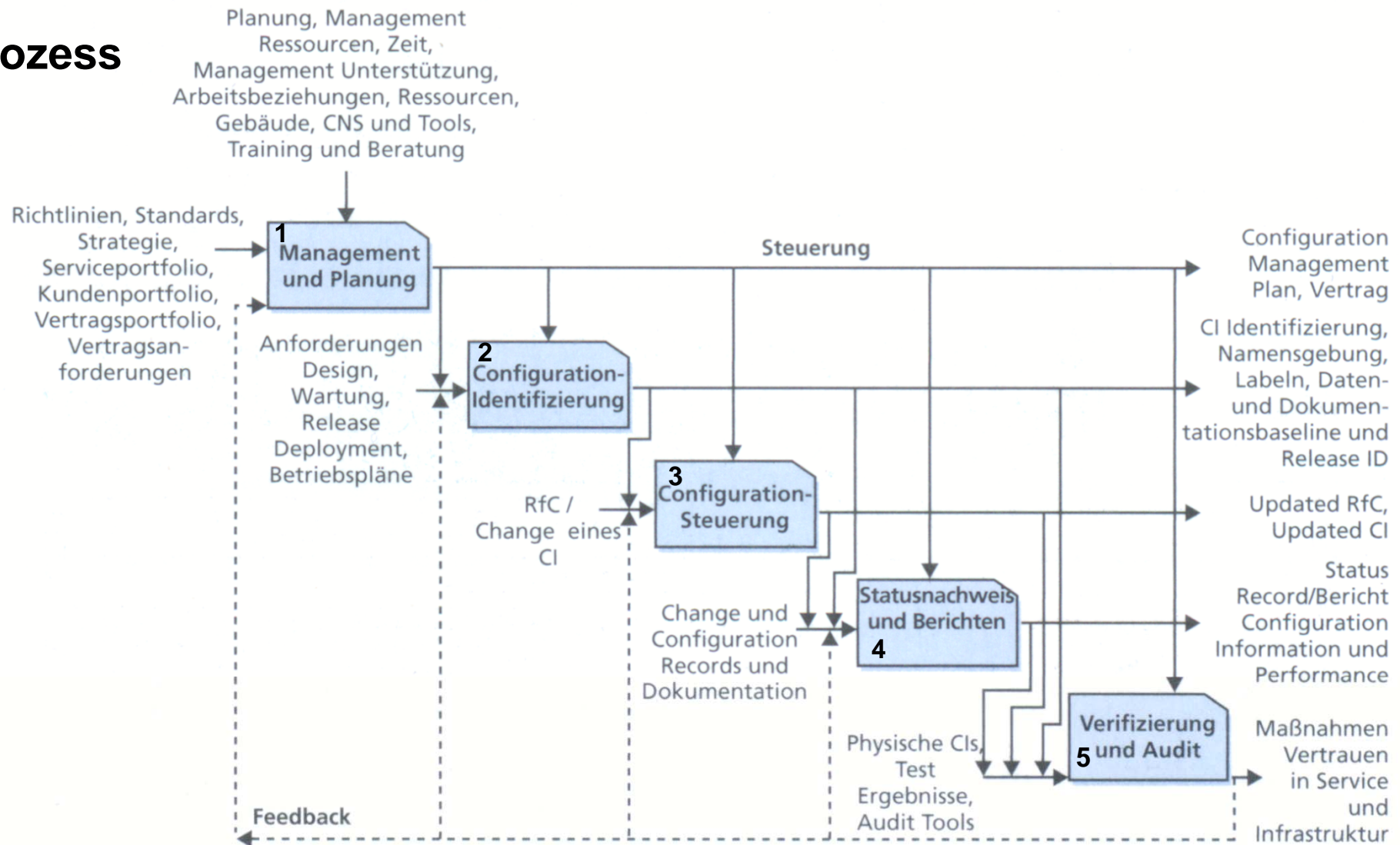
#### Weitere Grundbegriffe

- **Definitive Media Library (DML, Maßgebliche Medien Bibliothek)**
  - ◆ Sicherer Aufbewahrungsort an dem alle **autorisierte Versionen** (Master Copy) aller Medien CIs aufbewahrt und geschützt werden
    - Z.B. Originalversionen gekaufter Software inkl. Lizenzdokument
    - Z.B. Originalversionen selbst entwickelter Software (Quell- und Programmcode)
  - ◆ Sollte von Produktions-, Test- und Entwicklungsumgebung getrennt sein
  
- **Definitive Hardware Store (DHS, Maßgebliches Hardware Lager)**
  - ◆ Aufbewahrungsort von Ersatzteilen und Hardwarekomponenten insb. von standardisierten Grund- oder Basiskonfigurationen
  - ◆ Zum Austausch oder Reparatur von entsprechenden Konfigurationen der IT-Infrastruktur

## 3. Service Transition

### 3.3 Asset&Configuration Management

#### Prozess



Quelle: ITSF-V3-TB, S.120

# Configuration Management Prozess

## 1. Management und Planung

- Erstellung von Policies
  - ◆ Scoping
  - ◆ Namenskonventionen
  - ◆ Updateregeln
- Festlegung der Detaillierungstiefe (Scope, CI-Breakdownlevel)
  - ◆ CIs nur soweit herunter brechen wie sie eigenständig installiert oder geändert werden können
- Inventur: Analyse vorhandener Informationen

## 2. Configuration Identifizierung

- Erstellung Datenmodell
- Festlegen der CI-Attribute und des CI-Owner über Lebenszyklus  
z.B. in RACI-Matrix
- Erfassung (Inventur) der CI in CMS, CMDB, DHS, DML
- Erfassung der CI-Beziehungen



# Configuration Management Prozess

## 2. Configuration Identifizierung (Forts.)

### RACI-Matrix

- Responsible: Durchführungsverantwortung
- Accountable: Rechenschaftspflichtig, Kostenverantwortung (Disziplinarisch, Kosten)
- Consulted: hinzugezogen, Fachverantwortung
- Informed: zu informieren, Informationsrecht

#### Beispiel

Autoreparatur

	Service-Manager	Meister	Geselle	Lagerist	Geschäftsführer
Termin mit Kunden vereinbaren	R	C			A
Fahrzeug untersuchen		R	I	I	A
Ersatzteile beschaffen		A		R	
Reparatur durchführen	I	A	R		
Rechnung erstellen	R		C		A

Service-lebenszyklus-Phase	Beispiel von betroffenen Serviceebenszyklus-Assets und -CIs	Service Strategy	Service Design	Service Transition	Service Operation	CSI
Service Strategy	Portfolios – Service-Vertrag, Kunde Service-Strategie-Anforderungen Servicelebenszyklus-Modell	A	C	C	R	C
Service Design	Service Package (einschließlich SLA) Service Design Package, z. B. Service-Modell, Vertrag Service Management-Plan des Lieferanten Prozess-Schnittstellen-Definition Plan zur Einbindung des Kunden Release-Grundsätze Definition von Release-Paketen	I	A	C	R	C
Service Transition	Service Transition-Modell Testplan Kontrollierte Umgebungen Build-/Installationsplan Buildspezifikation Releaseplan Deploymentplan CMS SKMS Release Package Release-Baseline Release-Dokumentation Bewertungsbericht Testbericht	I	C	A	R	C
Service Operations	Service Operations-Modell Service Support-Modell Service Desk Anwender-Assets Anwender-Dokumentation Operations-Dokumentation Support-Dokumentation	I	C	C	A/R	R
Continual Service Improvement	CSI-Modell Service Improvement Plan Service Reporting-Prozess	C	C	C	A	R

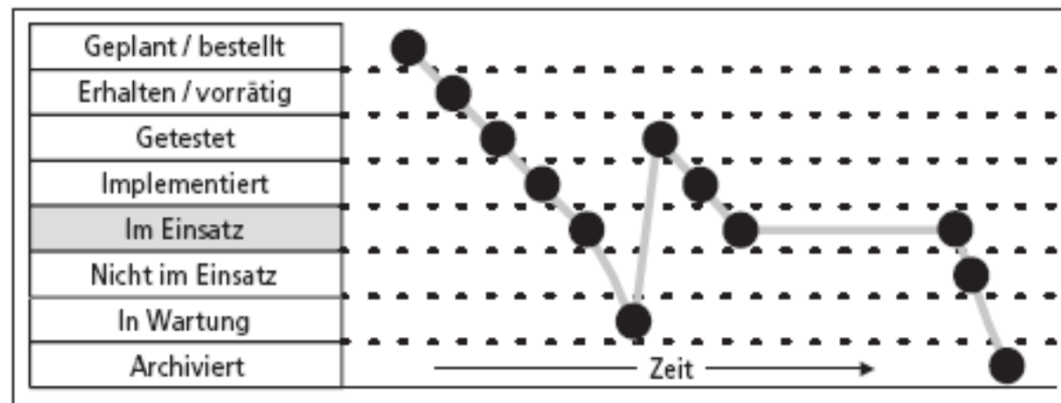
# Configuration Management Prozess

## 3. Steuerung (control)

- Sicherstellen der Aktualität der CMDB
- Sicherstellen, dass nur zugelassene und identifizierte CIs im Einsatz sind
- Sicherstellen, dass CI nur mit entsprechender Dokumentation CIs hinzugefügt, angepasst, ersetzt oder entfernt werden (z.B. in Form von genehmigten RFC)

## 4. Statusnachweis und Reporting

- Speicherung aktueller und historischer Daten über den Status eines CI im Laufe dessen Lebenszyklus
- Statusüberwachung und Verfolgung der Statusänderungen



Quelle: ITSF V2

## 5. Verifizierung und Audit

- Auditoren überprüfen die Inhalte des CMS, ob
  - ◆ CIs noch existieren
  - ◆ Sind für alle Änderungen RfCs gestellt worden
  - ◆ Vollständigkeit und Aktualität der CMS, CMDB, DSL, DHS
  - ◆ Einhaltung der Namenskonventionen
  - ◆ Korrekte Handhabung von Varianten, Kopien und Lizenzen
  - ◆ Zeitpunkt
    - Nach Implementierung der CMDB bzw. eines CMS
    - Vor und/oder nach wichtigen Änderungen (z.B. Ausgliederung/Verkauf von Unternehmensteilen)
    - nach Krisen- oder Notfallsituation
    - nach einem Angriff (Hacking)

**Welcher der nachfolgend aufgeführten Begriffe gilt als Configuration Item (CI)?**

- A. Störung (Incident)**
- B. IT-Personal**
- C. Organigramm**
- D. Prozess**



**Welcher der nachfolgend aufgeführten Begriffe gilt als Configuration Item (CI)?**

- A. Störung (Incident)**
- B. IT-Personal**
- C. Organigramm**
- D. Prozess**

---

A. Falsch. Ein Incident ist kein CI.

B. Richtig. IT-Personal kann ein CI sein.

C. Falsch. Ein Organigramm ist zwar eine Dokumentation. Diese bezieht sich aber auf das Unternehmen und nicht auf Teile der IT-Infrastruktur.

D. Falsch. Ein Prozess ist kein CI, sondern die eine Folge von Aktivitäten.

Aber: Vgl. Buch V2 Abb. 6.1

Vgl. Buch V3 S. 259

CI's werden durch eine **Klassifikation** unterteilt, beispielsweise: Service, Hardware, Software, Dokumentation, **Mitarbeiter**.

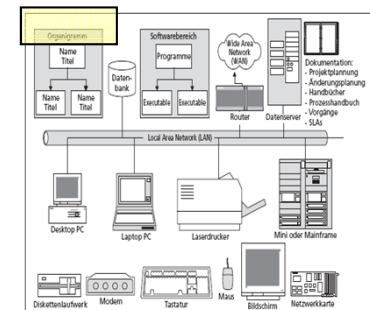


Abb. 6.1 Configuration Items

# Configuration Management Testfrage

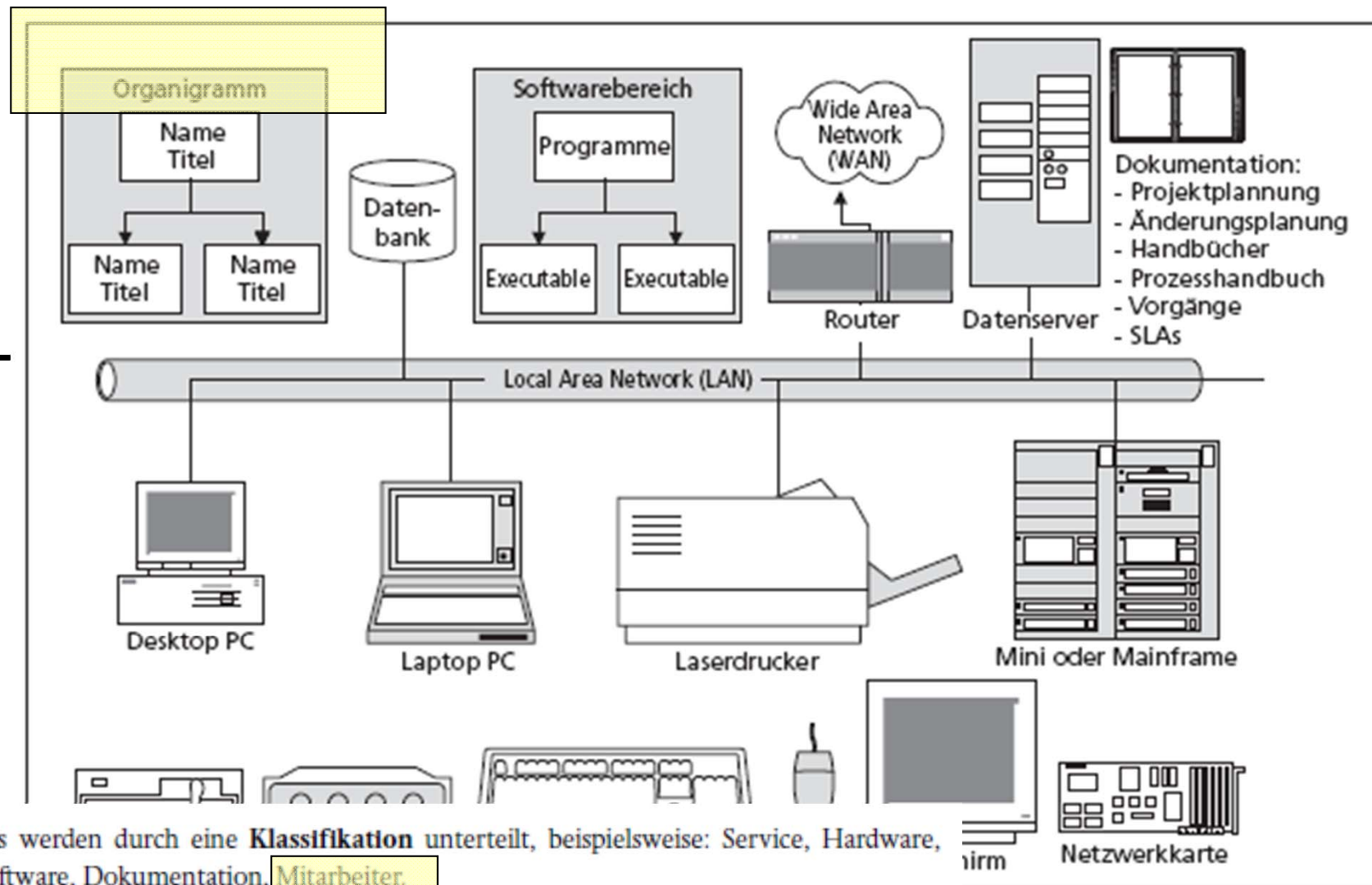
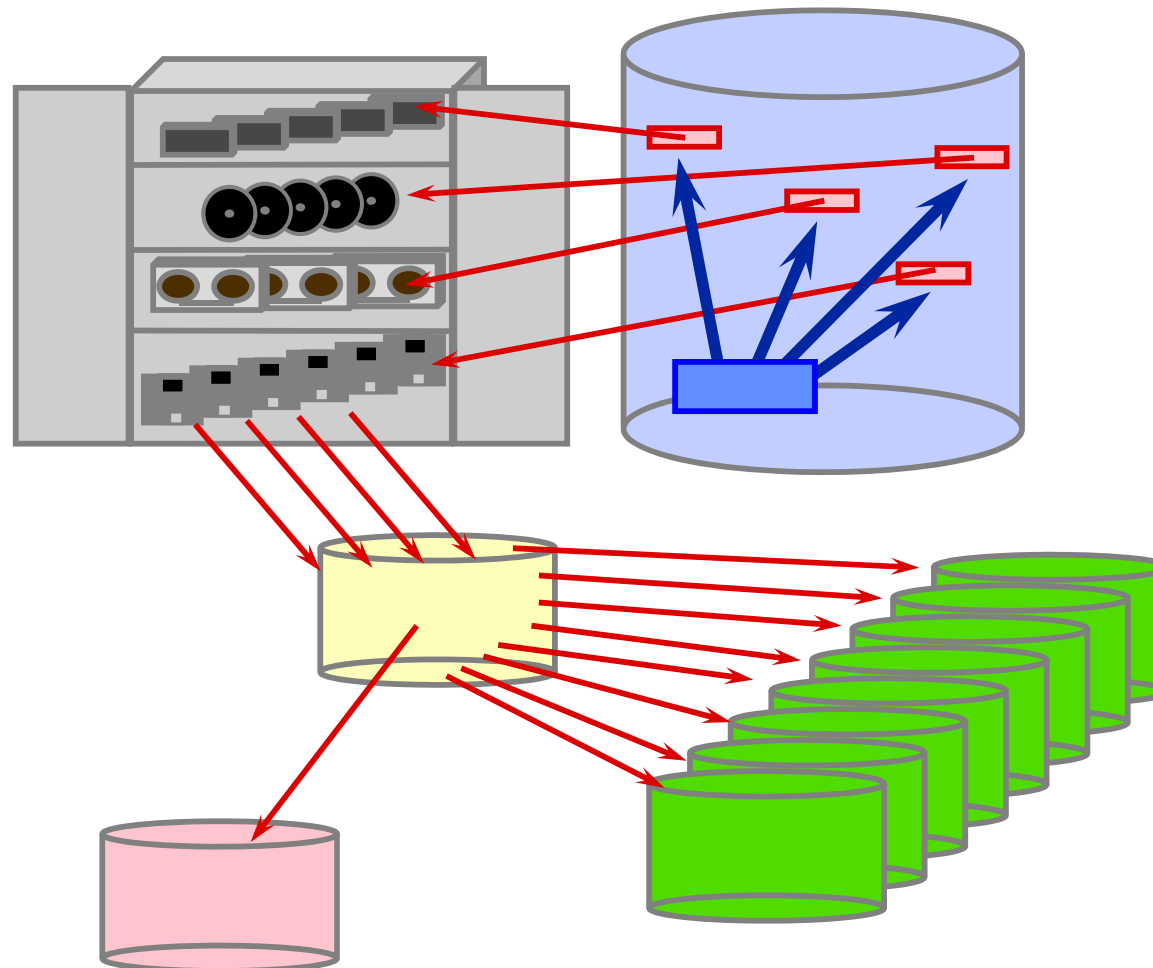


Abb. 6.1 Configuration-Items

## 3. Service Transition

### 3.4 Release&Deployment Management



## 3. Service Transition

### 3.4 Release&Deployment Management

#### Zielsetzung

- Erstellung, Testen und Bereitstellung bzw. Übergabe der Releases für die Produktionsumgebung gemäß den Design Packages
- Sicherstellung eines effizienten und risikoarmen Rollouts in die Produktionsumgebung
- Erstellung und Kommunikation entsprechender Release- und Deployment-Plänen

## 3. Service Transition

### 3.4 Release&Deployment Management

#### Grundbegriffe

##### ■ Release

Ein **Release** ist eine Menge neuer oder geänderter CIs die zusammen getestet und in die Produktionsumgebung ausgerollt werden.

##### ■ Release Unit

Komponenten eines IT Service, die im selben Release veröffentlicht werden und i.d.R eine in sich **abgeschlossene Funktion** darstellt.

Z.B. Notebook mit Software inkl. entspr. Lizenzen,

Remote Buchhaltungsanwendung inkl. kompletten Betrieb (DATEV)

##### ■ Release Package

Besteht aus einer oder mehreren Release Units, die zu einem Service gehören.

##### ■ Release Design

Unterscheidung nach der Methode des Rollouts eines Release Package:

- ◆ Big bang vs. Phasenorientiert (z.B. nach Wirtschaftsregionen, Funktionsumfang)
- ◆ Push vs. Pull (zentrale Verteilung vs. Download)
- ◆ Automatische vs. manuelle Verteilung

## 3. Service Transition

### 3.4 Release&Deployment Management

#### Grundbegriffe

##### ■ Releasearten

##### ◆ Major Release

Freigabe von neuer Hard- und/oder Software mit erheblicher Funktionalitätserweiterung.  
Zusätzlich werden Known Errors und Workarounds behoben.

##### ◆ Minor Release

- Freigabe mit geringfügigen Verbesserungen und Quick Fixes.
- Ein Minor Release dient als **Ausgangspunkt für Tests** “letzte funktionierende Basiskonfiguration”.

##### ◆ Emergency Release

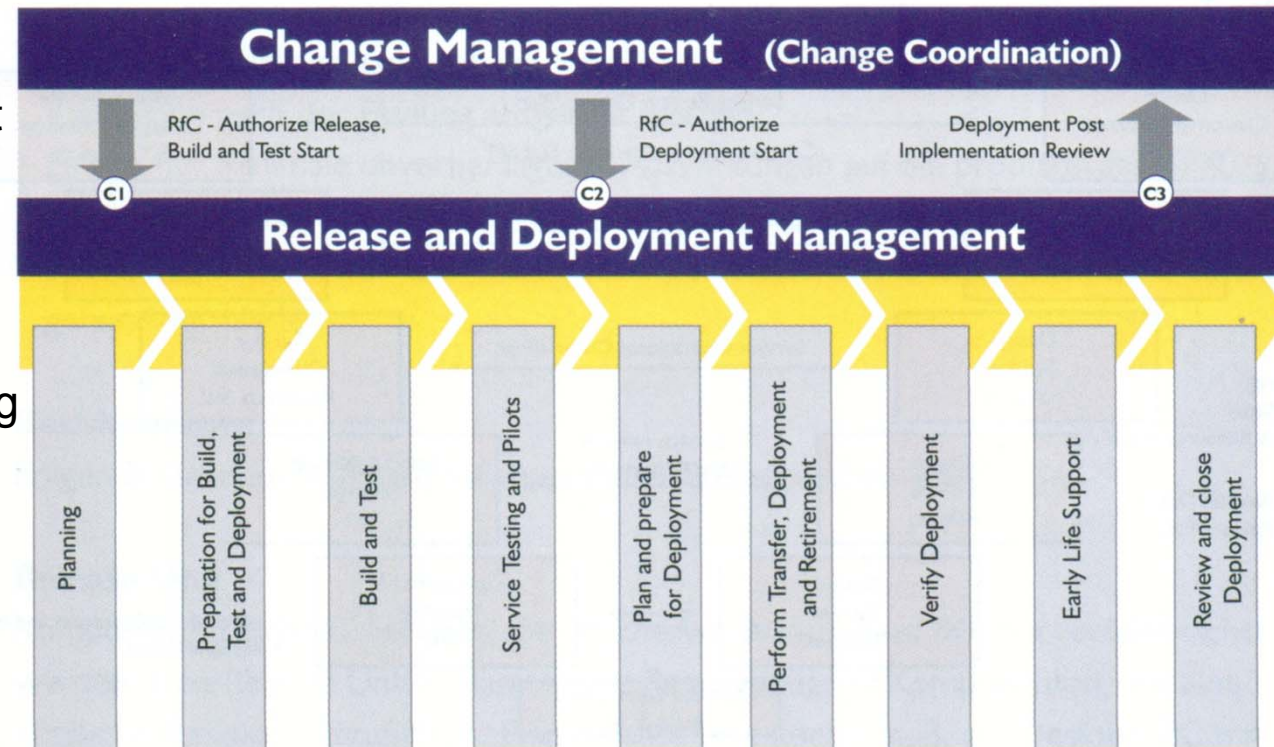
Dringende meist vorübergehende Behebung eines Known Errors.

## 3. Service Transition

### 3.4 Release&Deployment Management

#### Aktivitäten

1. Planung
2. Vorbereitung Build, Test u. Deployment
3. Build und Testen
4. Service-Tests und Pilotläufe
5. Planung u. Vorbereitung Deployment
6. Transfer, Deploym't u. Ausserkraftsetz'g
7. Deployment verifizieren u. bestätigen
8. Early Life Support
9. Review und Abschluss



Quelle: learnITIL

## 3. Service Transition

### 3.4 Release&Deployment Management

#### 1. Planung

- Erstellung der erforderlichen Rolloutpläne (z.B. gemäß **V-Modell**)
  - ◆ Build- und Testpläne
  - ◆ Schulungspläne
  - ◆ Backout/Fallback-Plan
- Freigabe der Pläne durch das Change Management

#### 2. Vorbereitung Build, Test und Deployment

- Abgleich des Designs mit den Anforderungen an die neuen/geänderten Services
- Freigabe für Build und Test

#### 3. Build und Testen

- Eigenentwicklung oder Kauf (Make or buy) der erforderlichen CIs
- Erfassung der CIs in CMDB
- Zusammenstellung eines testfähigen Release Packages (z.B. als Baseline)  
inkl. erforderlicher Lizenzen und Dokumentationen (Verfahren, Vorlagen, Handbücher)
- Übergabe an Testumgebung



## 3. Service Transition

### 3.4 Release&Deployment Management

## 4. Service-Tests und Pilotläufe

- Testmanagement: Planung und Koordination der Testaktivitäten
  - ◆ Vertiefung in 3.5 „Service-Validierung und –Test“

## 5. Planung u. Vorbereitung der Serviceverteilung (Deployment)

- Einschätzung, ob jedes Deployment-Team auf das Deployment vorbereitet ist (Readiness Assessment)
  - ◆ Sind diese für die Implementierung des Release-Pakets bereit?
  - ◆ Haben diese den Aktivitätsplan vorbereitet?
  - ◆ Sind die möglichen Risiken formuliert worden?
  - ◆ Ist jeder hinreichend geschult?
  - ◆ Gab es kurzfristige Änderungen in den Spezifikationen?

## 3. Service Transition

### 3.4 Release&Deployment Management

#### 6. Transfer, Deployment u. Ausserkraftsetzung

- Wichtige Aktivitäten
  - ◆ Zahlung Support- und Wartungskosten an Dritte
  - ◆ Überführung des Business und der Organisation (erforderliche organisatorische Änderungen)
  - ◆ Veröffentlichung von Dokumentation (erforderliche Richtlinien, Verfahrensanweisungen, Handbücher)
  - ◆ Übergabe von Service Management-Ressourcen (Übergabe an die verantwortlichen Teams)
  - ◆ Transfer des Service (Analyse der Service-Leistungsfähigkeit, Konfigurations-Audits, Aktualisierung des Service-Katalogs)
  - ◆ Deployment des Service - alle für die Verbreitung und Installation des Service notwendigen Aktivitäten
  - ◆ Beendigung redundanter Services - redundante Services werden eingestellt
  - ◆ Entfernung überflüssiger Assets

#### 7. Deployment verifizieren und bestätigen

- Bestätigen, dass der Service wie geplant genutzt werden kann
- Nach Abschluss aller Deploymentaktivitäten

#### 8. Early Life Support

- Zusätzliche Unterstützung nach dem Deployment während der Stabilisierungsphase

#### 9. Review und Abschluss



## 3. Service Transition

### 3.4 Release&Deployment Management

**Welcher Prozess übernimmt die Koordination über die Verteilung eines neuen Software-Releases?**

- A. Change Management
- B. Configuration Management
- C. Release Management
- D. Service Level Management

## 3. Service Transition

### 3.4 Release&Deployment Management

Welcher Prozess übernimmt die Koordination über die Verteilung eines neuen Software-Releases?

- A. Change Management
  - B. Configuration Management
  - C. Release Management
  - D. Service Level Management
- 

- A. Richtig. Kontrolle und Leitung (d.h. Koordination) der Verteilung von Releases finden unter der Verantwortung des Change Managements statt.
- B. Falsch. Das Configuration Management führt keine Regie über die Verteilung von Releases, ist aber an der Lieferung von Informationen über Configuration Items (CIs) und deren Änderung hinsichtlich der Verteilung betroffen.
- C. Falsch. Das Release Management führt keine Regie über die Verteilung von Releases, sondern liefert sachliche Informationen über Releases und die Planung.
- D. Falsch. Das Service Level Management führt keine Regie über die Verteilung von Releases, sondern liefert den Anwendern Berichterstattung über die Releases.

## 3. Service Transition

### 3.4 Release&Deployment Management

**An der Implementierung der neuen Version einer Anwendung sind sowohl das Change Management als auch das Release Management beteiligt.**

**Für welche der unten aufgeführten Aufgaben ist der Prozess Change Management in diesem Zusammenhang zuständig?**

- A.** Das Change Management ist in dieser Phase mit der Ausführung betraut.
- B.** Das Change Management spielt in dieser Phase eine koordinierende Rolle.
- C.** Dem Change Management obliegt die Prüfung, ob die neue Anwendung ordnungsgemäß funktioniert.
- D.** Das Change Management erstellt hierfür einen Request for Change (RFC).

## 3. Service Transition

### 3.4 Release&Deployment Management

**An der Implementierung der neuen Version einer Anwendung sind sowohl das Change Management als auch das Release Management beteiligt.**

**Für welche der unten aufgeführten Aufgaben ist der Prozess Change Management in diesem Zusammenhang zuständig?**

- A. Das Change Management ist in dieser Phase mit der Ausführung betraut.
  - B. Das Change Management spielt in dieser Phase eine koordinierende Rolle.
  - C. Dem Change Management obliegt die Prüfung, ob die neue Anwendung ordnungsgemäß funktioniert.
  - D. Das Change Management erstellt hierfür einen Request for Change (RFC).**
- 
- A. Falsch. Die Ausführung liegt beim Release Management, die Koordination beim Change Management (vgl. Testfrage 1).
  - B. Richtig. Das Change Management koordiniert die Entwicklung, Test, Implementiert und Funktionsprüfung/Abnahme (vgl. Change Mgmt „Grau/Weis-Folie“).
  - C. Falsch. Die Prüfung der Funktionsfähigkeit wird durch das Change Management koordiniert und nicht durchgeführt.
  - D. Falsch. RFC erstellt das Problem Management.

## 3. Service Transition

### 3.5 Service-Validierung und Test

#### Zielsetzung

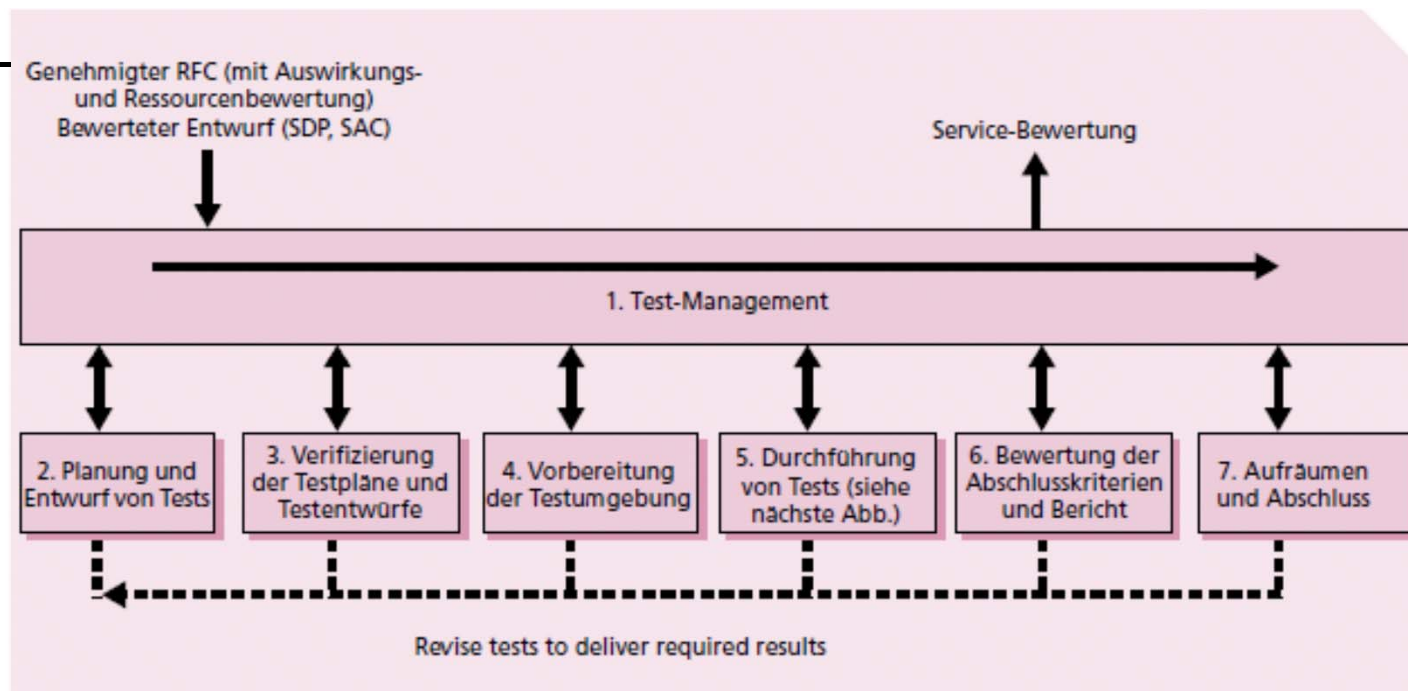
- Sicherstellen, dass das Release Package die **Kundenanforderungen** abdeckt
- Qualitätssicherung der vom Service Design an das Release&Deployment Management gelieferten neuen oder geänderten Services
- Service-Validierung und Test soll damit sicher stellen, dass neue oder geänderte Services
  - ◆ zweckmäßig (fit for purpose, utility) und
  - ◆ einsatzbereit (fit for use, warranty) sind
  - ➔ einen Mehrwert liefern

## 3. Service Transition

### 3.5 Service-Validierung und Test

#### Prozessaktivitäten

1. Test-Management
2. Planen u. Entwerfen von Tests
3. Verifizierung Testpläne
4. Vorbereitung Testumgebung
5. Testdurchführung
6. Bewertung der Abschlusskriterien und Bericht
7. Clean up & Closure





## 3. Service Transition

### 3.5 Service-Validierung und Test

#### 1. Test-Management

- Planung, Steuerung und Kontrolle des gesamten Testprozesses inkl. Berichterstattung
  - ◆ Planung der Test-Ressourcen,
  - ◆ Priorisierung und Einplanung, was, wann getestet werden soll,
  - ◆ Behandeln von Incidents, Problemen, Fehlern, Risiken und offenen Punkten,
  - ◆ Überwachen, dass eingehende bekannte Fehler bearbeitet werden,
  - ◆ Beseitigen der erkannten Fehler über das Change Management, damit eine möglichst fehlerfreie Überführung in die Produktivumgebung erfolgen kann,
  - ◆ Überwachen des Arbeitsfortschritts und Zusammenführen der Rückmeldungen aus den Validierungs- und Testaktivitäten,
  - ◆ Erfassen der Configuration Baselines,
  - ◆ Erfassen, Analysieren, Auswerten und Behandeln der Testmessdaten.

#### 2. Planen u. Entwerfen von Tests

- Die Planungs- und Entwurfsaktivitäten beginnen früh im Servicelebenszyklus und beinhalten folgende Aspekte:
  - ◆ Ressourcen,
  - ◆ Hardware, Netzwerk, Anzahl benötigter Mitarbeiter, Qualifikationen etc. sowie die damit verbundenen Kapazitäten,
  - ◆ Planen der Meilensteine, Arbeitsergebnisse, Übergabe- und Abnahmetermine,
  - ◆ Ressourcen auf geschäftlicher, beziehungsweise Kundenseite,
  - ◆ vereinbarte Zeiträume zur Auswertung von Berichten und anderen Arbeitsergebnissen,
  - ◆ Budgetierung und Finanzierung. Abgleich des Designs

## 3. Service Transition

### 3.5 Service-Validierung und Test

#### 3. Verifizierung der Testpläne

- Mit dieser Prüfung soll sichergestellt werden, dass
  - ◆ die Test-Modelle die Risikoprofile des Service angemessen abdecken,
  - ◆ die wesentlichen Integrationsaspekte und Schnittstellen durch die Test-Modelle abgedeckt werden,
  - ◆ die Test-Skripte korrekt und vollständig sind

#### 4. Vorbereitung Testumgebung

- Zur Vorbereitung der Test-Umgebungen sollten Prozeduren zur Erstellung kontrollierter Test- und Auslieferungsumgebungen eingesetzt werden.
- Schaffen einer Baseline für die Testumgebung

#### 5. Testdurchführung

- Die Tests werden mittels manueller oder automatisierter Prozeduren durchgeführt.
- Die Tester müssen während des Tests alle Auffälligkeiten protokollieren.
- Falls ein Test scheitert, sollten die aufgetretenen Störungen oder Begebenheiten behoben oder dokumentiert werden.
- Nach der Behebung eines Fehlers wird ein entsprechender Nach-Test durchgeführt.

## 3. Service Transition

### 3.5 Service-Validierung und Test

#### 6. Bewertung der Abschlusskriterien (Exitkriterien) und Bericht

- Die aktuellen Ergebnisse werden mit den erwarteten Ergebnissen verglichen.
- Die Ergebnisse können bewertet werden nach dem Erfüllungsgrad, den geschäftlichen Risiken oder danach, ob sich geplante Werte (wie z. B. Kosten) geändert haben.
- Der Auswertungsbericht bezieht sich auf die Testmetriken und stellt die Testergebnisse in Summe dar.
- Beispiele für Exit-Kriterien sind:
  - ◆ alle definierten funktionalen Aspekte können seitens der Anwender korrekt ausgeführt werden,
  - ◆ der Service erfüllt die Qualitätsanforderungen,
  - ◆ die Configuration Baselines sind in der CMS erfasst.

#### 7. Clean up & Closure

- Zurücksetzen der Testumgebung
- Abschluss und Archivierung der erforderlichen Testdokumentation
- Lessons Learned

## 3. Service Transition

### 3.6 Evaluation

#### Zielsetzung

- Prozess, der für die **Bewertung** eines neuen oder geänderten IT Service verantwortlich ist. Eine Evaluierung bezeichnet den **Vergleich eines Ist-Ergebnisses mit dem beabsichtigten Plan-Ergebnis**  
z.B. Performanceverbesserung, Bestimmtes Preis-/Leistungsverhältnis
- Abgrenzung zu Validierung&Test:
  - ◆ V&T prüft gegen die fachlichen Kundenanforderungen.
  - ◆ Evaluation prüft gegen Zweck warum man den Service (gem. Kundenanforderungen) eingeführt hat
- Der Prozess wird in der Regel innerhalb des Change Management eingesetzt
- Es werden neue oder geänderte Services betrachtet,
  - ◆ die in der Phase Service Design definiert worden sind,
  - ◆ sich in der Phase Transition, aber vor der endgültigen Übergabe an Service Operation befinden.

## 3. Service Transition

### 3.6 Evaluation

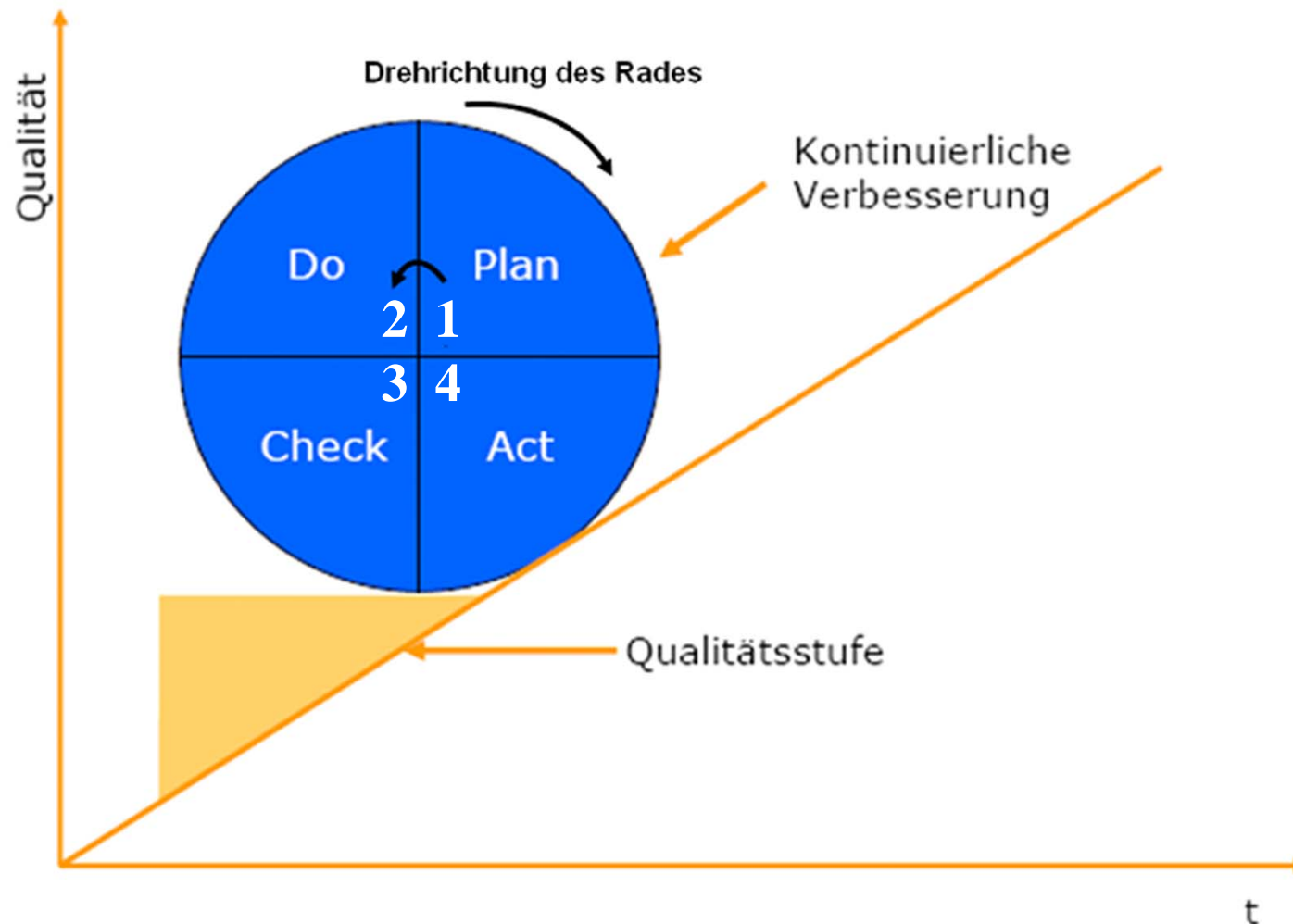
#### Grundsätze

- Evaluierung von Service Designs oder Serviceänderungen bevor sie in den operationalen Betrieb überführt werden
- Jede festgestellte Abweichung zwischen erwarteter und tatsächlicher Leistungsfähigkeit wird vom Kunden oder Kundenvertreter akzeptiert, abgelehnt oder zur Nachbesserung zurückgereicht
- Evaluierung findet stets unter Einbindung des Auftraggebers statt
- Der Evaluierungsprozess nutzt das Plan-Do-Check-Act-Modell (PDCA), um Konsistenz über alle Evaluierungen hinweg sicherzustellen

## 3. Service Transition

### 3.6 Evaluation

## PDCA Qualitätskreis von Deming



**Qualitätssteigerung**

**Qualitätssicherung**

- 1 Planen (Plan)
- 2 Ausführen (Do)
- 3 Messen (Check)
- 4 Anpassen (Act)

**Qualitätssorgfalt**

**Qualitätssystem**

Quelle: in Anlehnung an itSMF

## 3. Service Transition

### 3.7 Knowledge Management

#### Zielsetzung

- Verbesserung der Qualität der Entscheidungsprozesse (des Managements) durch Sicherstellung, dass verlässliche und gesicherte Informationen während des Service Lebenszyklus verfügbar sind.
- Gewährleistung, dass den Mitarbeitern des Service Providers alle notwendigen Informationen zur Verfügung stehen

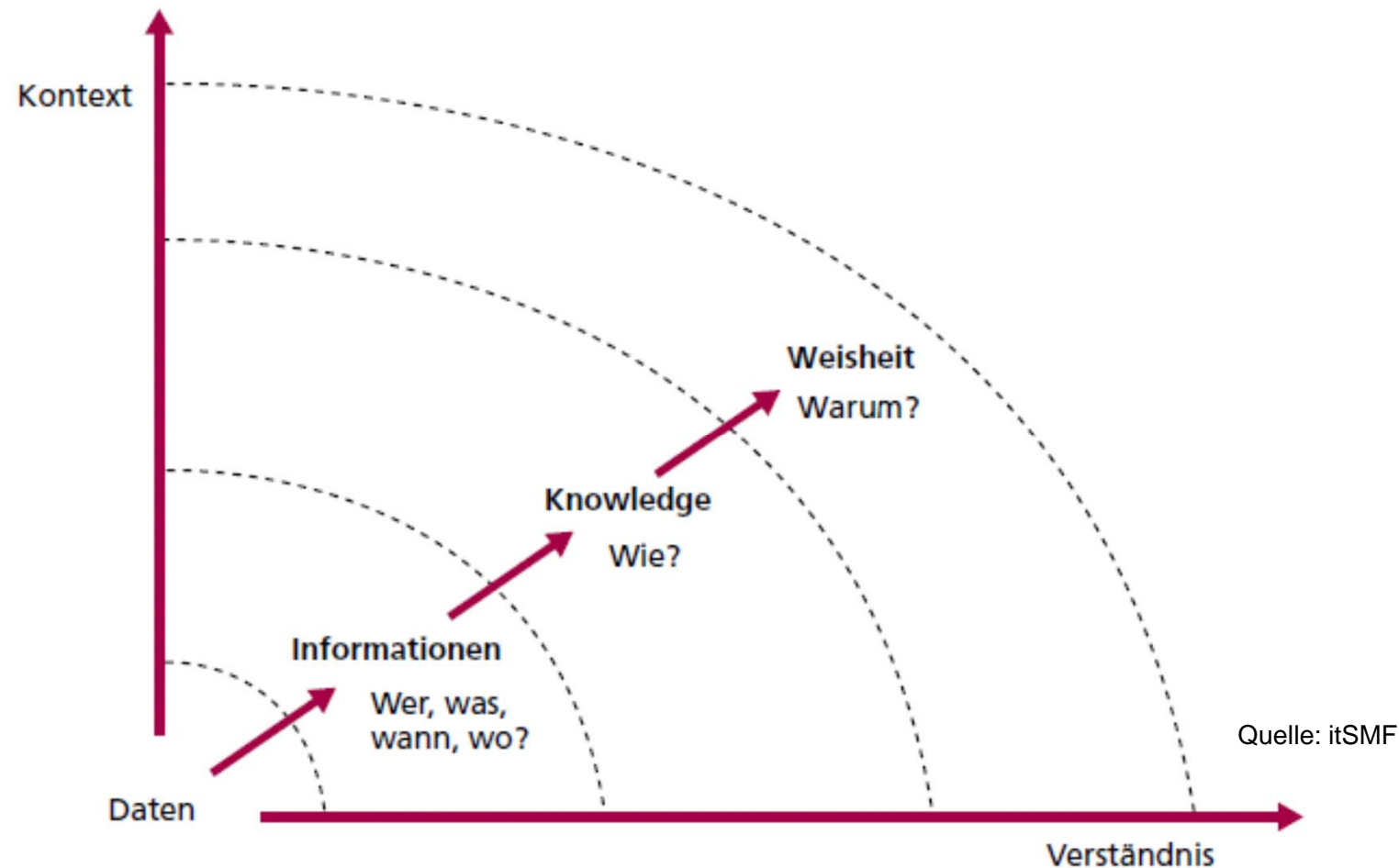


## 3. Service Transition

### 3.7 Knowledge Management

#### Grundkonzepte: DIKW-Modell

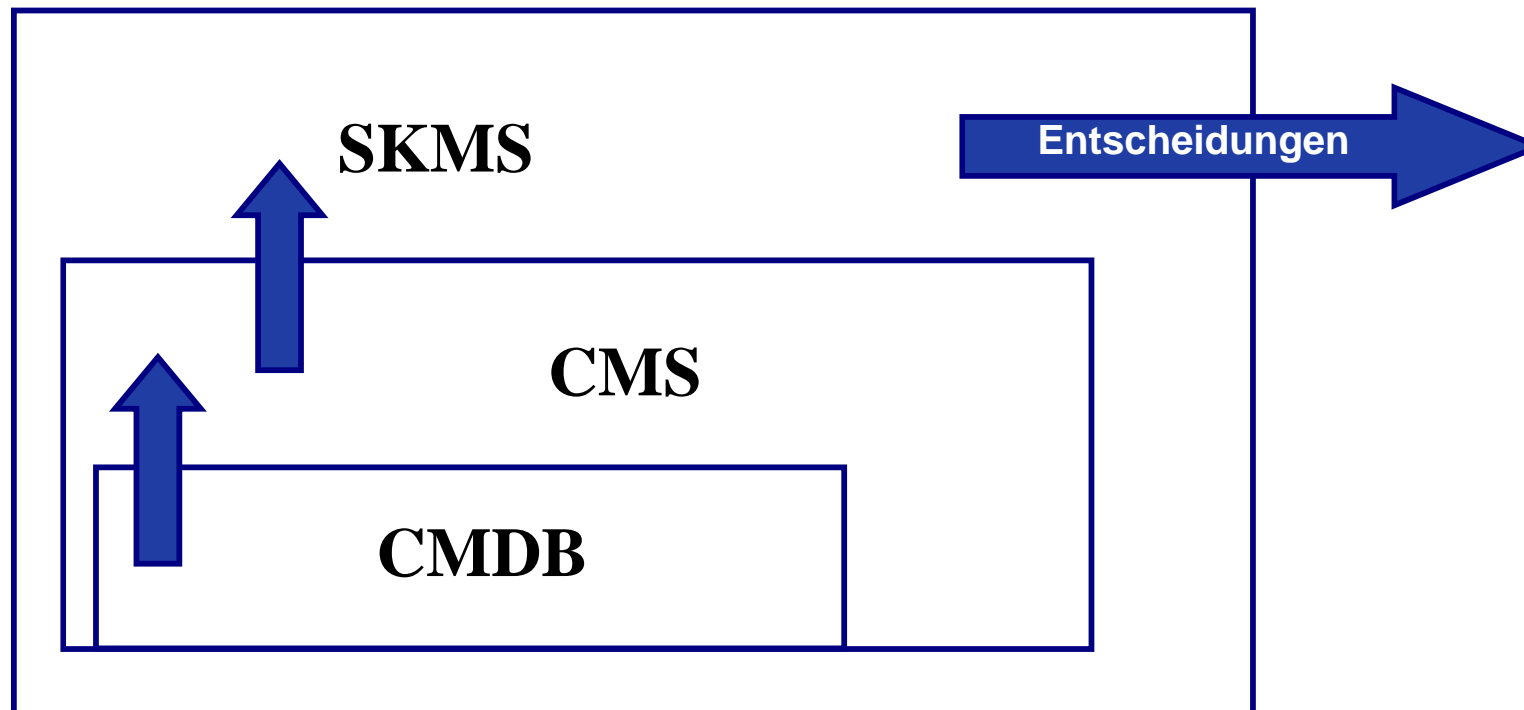
- Wissensmanagement wird oft durch Einsatz der DIKW-Struktur visualisiert: Daten-Informationen-Knowledge(Wissen)-Weisheit



### 3. Service Transition

#### 3.7 Knowledge Management

**Grundkonzepte: Service Knowledge Management System (SKMS)**



Quelle: OGC