

# ÄNDERUNGSABLAUF NACH ITIL

Armin Jakupovic, Roman Etter

AB05 – Modul 150

## Inhalt

ITIL Service Transition .....	2
Knowledge-Management .....	2
Service Asset & Configuration Management.....	2
Service-Validierung & Testing.....	2
Anwendungs-Entwicklung .....	2
Release & Deployment Management.....	2
Change Evaluierung.....	2
Projekt-Management (Transition Planning and Support).....	2
Change-Management.....	3
Reflexion .....	3
RfC Prozess.....	4
RFC Prozess im Ausbildungsbetrieb.....	4
Vorschlag für Verbesserungen am Prozess.....	5
Unterschiede Agile und Wasserfall Prozesse.....	6
Grundlegendes Vorgehen.....	6
Flexibilität für Änderungen.....	6
Prozess und Aufteilung.....	6
Fokus .....	6

## ITIL Service Transition

Die ITIL Service Transition gilt als eine ITIL Disziplin und Best Practice Vorgabe, die aus verschiedenen Prozessen besteht und den Change Prozess miteinbezieht. Ziel ist es, dass der Service Provider sicherstellt, dass die bereitgestellten und gelieferten Services den mit dem Service Abnehmer vereinbarten Anforderungen und Abmachungen übereinstimmen.

Ob die ITIL Disziplin erfüllt ist oder nicht kann unterschiedlich gemessen werden. Üblich ist, dass Kennzahlen wie die Anzahl an CAB-Meetings, Durchlaufzeit eines Changes, die Anzahl von Changes pro Priorität oder auch die Akzeptanzrate von RfC, verwendet werden.

## Knowledge-Management

Knowledge-Management ist ein einziger zentraler Prozess, der für alle IT-Service-Management-Prozesse Wissen zur Verfügung zu stellen soll. Der Knowledge-Management Prozess stellt sicher, dass alle Informationen im System abgelegt werden. Die Daten sind dabei konsistent und leicht wiederauffindbar.

## Service Asset & Configuration Management

Dieser Prozess ist dafür verantwortlich, Service- und Konfigurationsinhalte zu kontrollieren, administrieren und dokumentieren. Es ist ebenfalls für jegliche Beziehungen und Abhängigkeiten unter den einzelnen Elementen verantwortlich.

## Service-Validierung & Testing

Hier wird sichergestellt, dass neue Releases und deren Services qualitätsgeprüft werden. Mit diesem Prozess wird bestimmt, ob denn der IT-Betrieb überhaupt in der Lage ist, den neuen Service angemessen zu unterstützen.

## Anwendungs-Entwicklung

Die Anwendungs-Entwicklung ist natürlich dafür zuständig, die Anwendungen und Systeme und deren erforderlichen Funktionalitäten zu entwickeln. Zum Prozess gehört aber nicht nur das Entwickeln, sondern auch die Wartung von spezifischen Anwendungen der Kunden, wie auch die generelle Anpassung der Standardsoftware.

## Release & Deployment Management

Das Release & Deploy Management schaut, dass die Live-Umgebung geschützt wird und nur geprüfte Komponenten deployt werden. Zudem planen sie, legen fest und kontrollieren wie ein Release getestet und ausgerollt wird.

## Change Evaluierung

Die Change Evaluierung beurteilt bedeutende Changes, wie der Implementierung eines neuen Service oder ein substanzieller Change an einem bestehenden Service. Das Resultat was bei dieser Beurteilung herauskommt, sagt aus, ob der Change zum nächsten Schritt vom Lebenszyklus gehen kann.

## Projekt-Management (Transition Planning and Support)

Das Projekt-Management soll die benötigten Ressourcen für einen Release Planen und Koordinieren. Dadurch soll der Release innerhalb des gesetzten Zeit- und Kostenrahmens mit entsprechender Qualität veröffentlicht werden können.

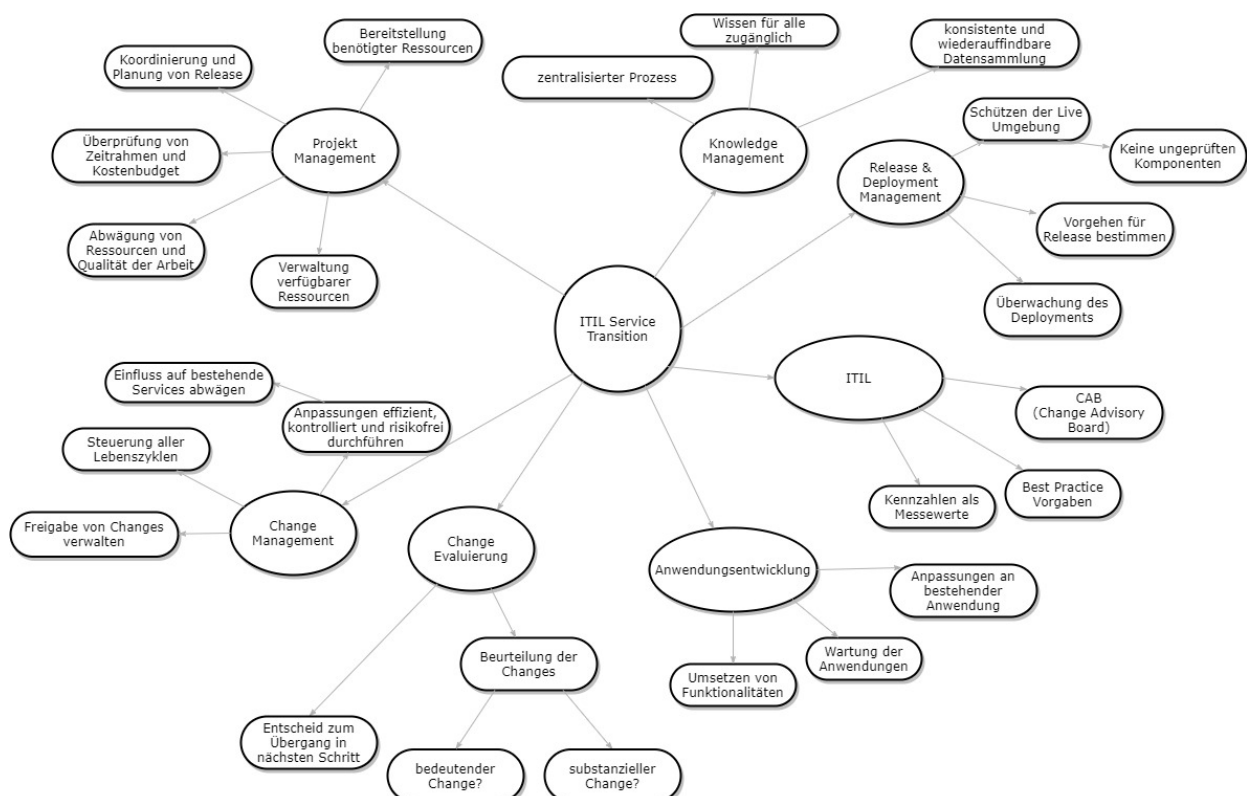
## Change-Management

Das Change-Management ist ein Prozess, der das Ziel hat, alle Anpassungen der IT-Infrastruktur kontrolliert, effizient und mit so wenig wie möglich Risiken für die bestehenden Services durchzuführen. Er steuert den Lebenszyklus aller Changes und überwacht diese. Der Prozess hat als Main Goal, nutzbare Changes zu erlauben und dabei negativen Nebenwirkungen aus dem Weg zu gehen, die die Funktionalität anderer Services beeinträchtigen. Das Change-Management ist sehr gross, es umfasst folgende Teil Prozesse:

- Change Management Support
- Bewertung von Requests for Change
- RFC Erfassung und Review
- Bewertung und Implementierung von Notfall-Changes
- Change Bewertung durch Change Manager
- Change Bewertung durch Change Advisory Board (CAB)
- Change Planung und Freigabe für Umsetzungsphase
- Freigabe der Deployment Phase
- Implementierung von Minor Changes
- Implementations-Review und Change-Abschluss

## Reflexion

Wir konnten im Rahmen dieses Arbeitsblattes ITIL noch näher kennenlernen. Wir hatten in den vorherigen Arbeitsblättern zwar ITIL Bezug gehabt und uns leicht vertraut damit gemacht, jedoch konnten wir uns hier darin vertiefen. Für beide war der «Request for Change» schon bekannt und klar. Wir hatten uns sowohl selbst als auch mit unserer Ansprechperson im Detail darüber unterhalten und dazulernen. Hingegen war uns ITIL Service Transition bis anhin noch kein bekannter Begriff. Wir haben aber schnell erkannt, welche Bedeutung diese Disziplin hat und wie wichtig sie eigentlich ist. Nachdem wir uns eine Weile damit auseinandergesetzt haben, erkannt wir auch, wo die Risiken liegen, wenn die Disziplin im grösseren Stil missachtet wird – Konsequenzen sowohl finanziell als auch kundenspezifisch.



## RfC Prozess

### RFC Prozess im Ausbildungsbetrieb

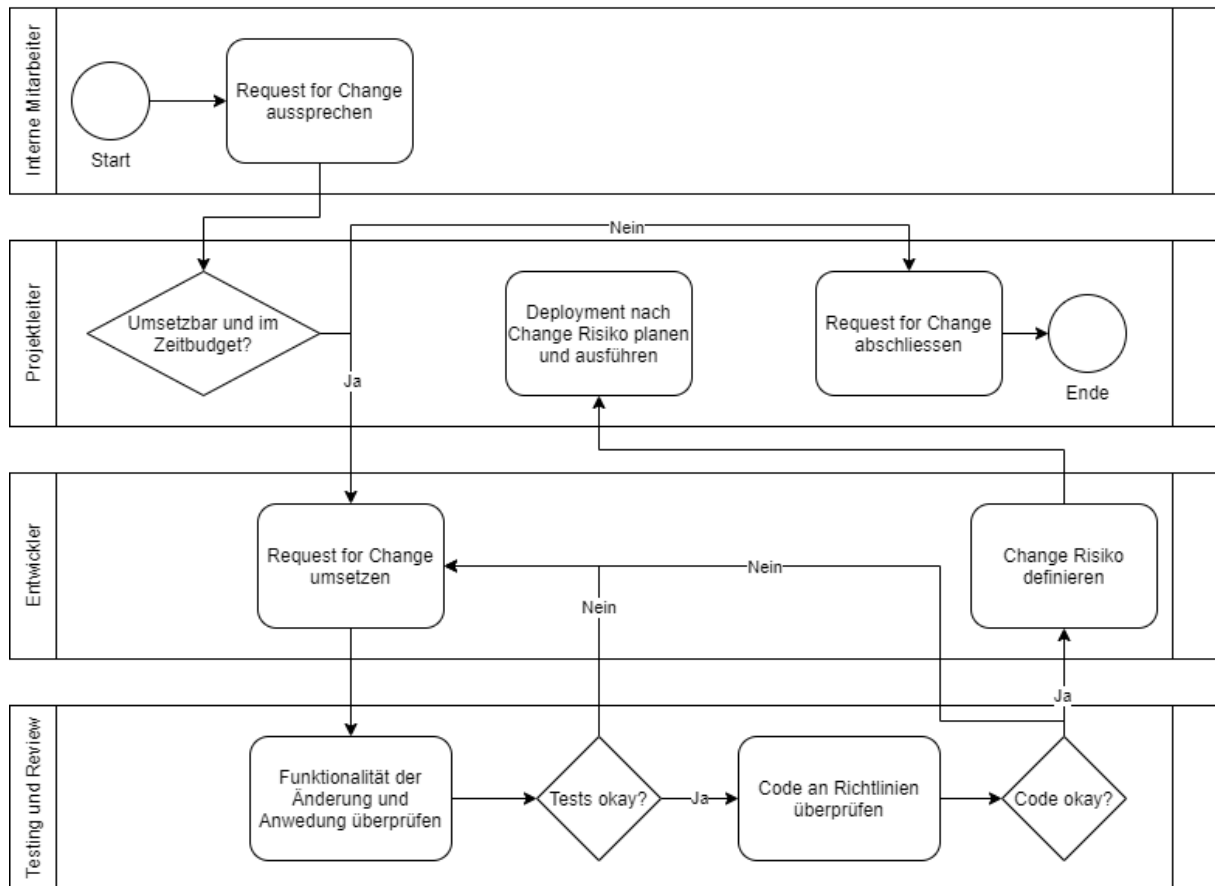


Abbildung 1 Aktueller RFC Prozess im Ausbildungsbetrieb

## Vorschlag für Verbesserungen am Prozess

Um den Prozess besser an ITIL anzupassen haben wir folgendes verändert.

- Die Entwickler sind nun auch am Planen des Changes beteiligt
  - Dies ist wichtig, da dadurch die Komplexität und Aufwand besser geplant werden kann
- Der Projektleiter, der die Rolle vom Change-Management hat, muss den Change nochmals autorisieren
  - Im bisherigen Ablauf hatte das Change Management zu wenig Kontrolle über den Lebenszyklus des Changes. Durch diese Änderung wird dem Change Management wieder mehr Kontrolle übergeben.
- Change evaluieren → besserer Begriff gewählt, um den Prozess zu beschreiben
- Das Deployment vor dem Release nochmals überprüft bzw. reviewt
  - Diese Änderung verfolgt ein ähnliches Ziel wie dieses vom Change autorisieren. Der Prozess des Deployments wird nachvollziehbarer dokumentiert und erläutert vorbeugende Prozesse. Zudem behandelt es die Aufgaben des Deployment Managements.

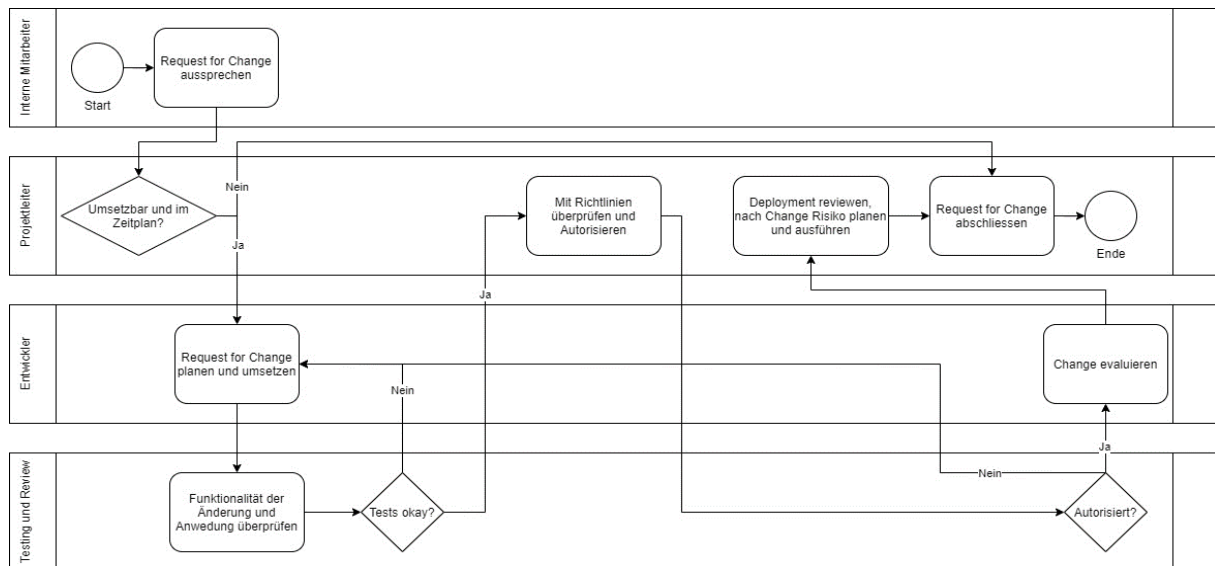


Abbildung 2 RFC nach Verbesserungsvorschlag

## Unterschiede Agile und Wasserfall Prozesse

Die Unterschiede zwischen Agil und Wasserfall sind in einigen Fällen enorm. Es ist daher wichtig, dass sich die Projektleitung für das Projekt am besten passenden Vorgehen entscheidet.

### Grundlegendes Vorgehen

Agile Entwicklungsumgebungen basieren auf einem inkrementellen Ansatz und erlauben für mehr Flexibilität. Projekte basierend auf Wasserfall hingegen setzen auf einen sequenziellen Entwurfsprozess. Das bedeutet, dass die einzelnen Schritte nacheinander abgeschlossen werden.

### Flexibilität für Änderungen

Der Agile Prozess ist dafür bekannt, sowohl dem Kunden als auch den Entwicklern viel Spielraum und Flexibilität zu bieten. Ein Kunde kann auch nach Abschluss der anfänglichen Planung eine Änderung an den Anforderungen erfragen kann. Diese können in der Regel problemlos aufgenommen und später umgesetzt werden. Der Prozess ist auch explizit für dieses Szenario konzipiert und erlaubt das Einführen von Änderungen zu jeder Zeit.

Der Wasserfall Prozess hingegen ist so ziemlich das genaue Gegenteil von jenem Agilen. Alle Anforderungen werden in der Planungsphase gesetzt und in der Umsetzungsphase implementiert. Änderungen an den Anforderungen können in der Regel nicht oder nur sehr schwer einbezogen werden, wenn die Planungsphase zu Ende ist. Das Modell selbst sieht Änderungen nach einer abgeschlossenen Planungsphase nicht vor und ratet davon ab. Vom Kunden werden von Anfang an klare Anforderungen erwartet.

### Prozess und Aufteilung

Der Prozess einer Agil Umgebung ist iterativ und in sogenannte «Sprints» unterteilt. Das bedeutet, dass Planung, Umsetzung und Testing sich bis zum Abschluss des Projektes wiederholt. Das Projekt als ganzes wird prinzipiell wie in kleinere Projekte unterteilt, die dann in einem vorgegebenen Zeitraum abgeschlossen werden – dies stellt den Sprint dar. Jedes unterteilte Projekt bzw. jeder Sprint beinhaltet die Planung, Umsetzung und das Testing.

Bei einer Wasserfall Umgebung sieht das ganze anders aus. Der Prozess ist, wie bereits erwähnt, sequenziell. Das Planen, Testen und Umsetzen verfolgt einen fixen Ablauf und wiederholt sich grundsätzlich nicht. So gibt es hier keine «Sprints» sondern Phasen, die klar von einem Zeit- und Ablaufplan definiert und gesteuert werden. Die Entwicklungsphase sieht vor, dass alle Komponenten fertig entwickelt werden. Testing während der Entwicklungsphase ist kein Thema, für dieses gibt es eine separate Phase, die nach der Entwicklungsphase folgt.

### Fokus

Der Agil Prozess hat einen erhöhten Fokus auf den Kunden, seine Bedürfnisse und Anforderungen. Der Input des Kunden spielt eine grosse Rolle und wird fortlaufend eingeholt, unabhängig von Phase und Sprint. Oftmals werden Besprechungen mit dem Kunden durchgeführt, nachdem ein Sprint erfolgreich abgeschlossen ist, um Feedback zu erhalten.

Der Wasserfallprozess hat einen deutlich schwächeren Bezug zum Kunden. Es bevorzugt eine Denkweise, die das erfolgreiche Abschliessen und Ausliefern der Anwendung in den Fokus rückt und entsprechend priorisiert. Der Kunde wird üblicherweise nur am Anfang in der Planungsphase einbezogen. Die restlichen Phasen bauen auf dem erlangten Wissen in der Planungsphase, beziehen den Kunden jedoch nicht mehr ein.