

## Change Management

Inf2020d, Sashauna Wray, Valentin Ehinger

### Änderungs-Prozess



*RFC für SBB nach Wasserfall-Methode.*

Im Änderungs-Prozess ist der Chaner dafür verantwortlich, dass die Veränderungen den laufenden Betrieb so gering wie möglich beeinflussen. Es werden auch die Aufträge von externen Providern geregelt. Der Changer stellt auch sicher, dass die Veränderungen auf der Produktiven Umgebung auch wirklich funktionieren. Im Änderung-Prozess existieren 8 Phasen. Sobald eine Phase erfolgreich abgeschlossen wurde und vom Changer überprüft wurde, kann mit der nächsten Phase angefangen werden.

#### Draft

Hier werden Ideen/ gewünschte Änderungen geprüft und abgewogen. Anschliessen wird der Change erfasst. Dies beinhaltet was gemacht werden muss, bis wann es fertig sein soll, welche Services betroffen sind und wer involviert sein wird.

#### Access

In diesem Abschnitt werden die Konkreten Tasks definiert. Der Change Koordinator/in legt dabei fest welcher Task von welchen Providern oder Gruppe bearbeitet wird.

#### Approve

Von dem Change Master oder Change Koordinator/in muss die bisherigen schritte überprüfen und anschliessend genehmigen, damit es in die nächste Phase übergehen kann.

### **Build and Test**

Alle Tasks werden auf einer Testversion umgesetzt und genau geprüft. Ein Task kann dabei mehrmals hin und her gehen. Erst wenn die Testphase erfolgreich abgeschlossen ist, geht es weiter zu Deploy.

### **Deploy**

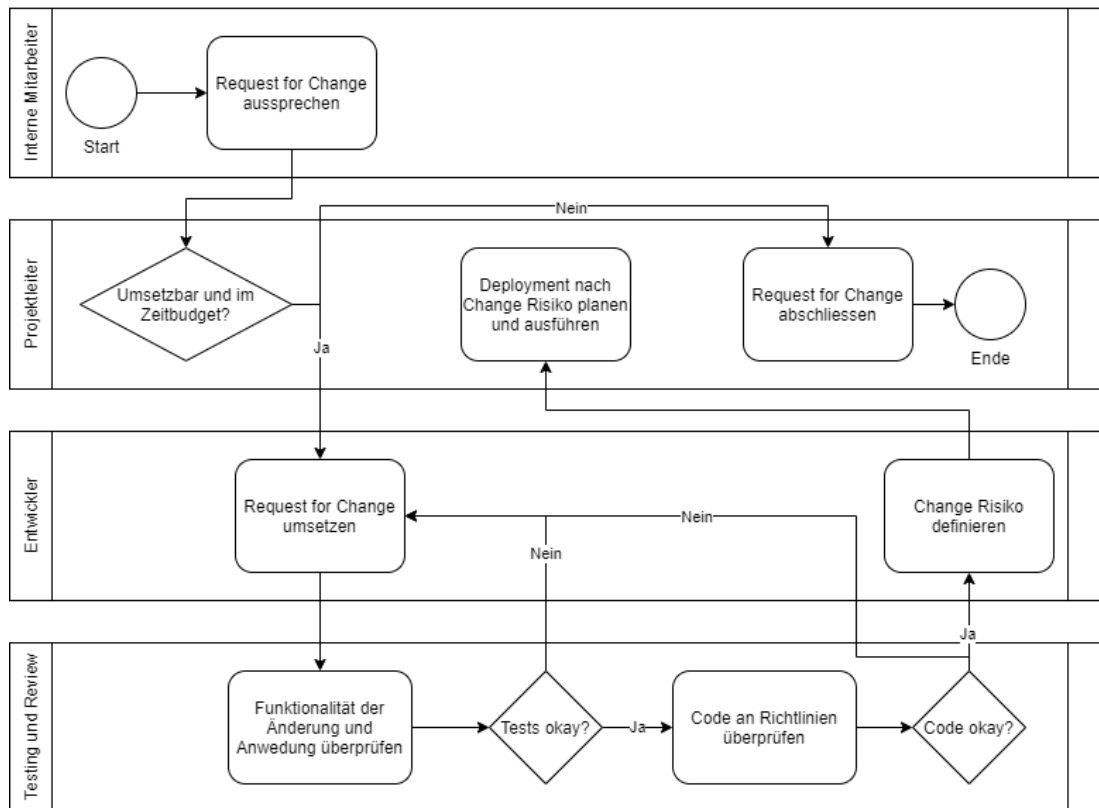
Auf diese Weise wird sichergestellt, dass die Veränderungen den Betrieb sehr unmerklich oder überhaupt nicht vom laufenden Betrieb gemerkt werden.

### **Review**

Nun wird bewertet, ob die Umsetzung gut gelungen ist. Diese Bewertung ist wichtig für den Change Master. So kann er den Prozess kontinuierlich verbessern. In einem letzten Schritt wird die Knowledge-Base an die Veränderungen angepasst. Mit dem kann bei Problemen schnell und zuverlässig geholfen werden.

### **Close**

Nachdem all diese Schritte ausgeführt wurden, ist der Änderungs-Prozess komplett abgeschlossen.



*RFC von interner InterDiscount-Dokumentation mit Ausbildner erstellt.*

Der hier präsentierte RFC ist auf einen normalen Change angewendet. Das heisst, dass ein Change-Vorschlag zuerst geprüft und autorisiert werden muss. Bei Interdiscount läuft ein RFC nach der Agilen Methode und nicht nach der Wasserfall Methode. Die Wasserfall Methode läuft sehr ähnlich ab. Bei Interdiscount werden die Risiken in der Planungsphase abgeschätzt. So können Risiken früh erkannt und mögliche Konflikte mit anderen Applikationen minimiert werden. Nach der Umsetzung werden die Risiken erneut geschätzt und festgehalten. Um Risiken so gut als möglich zu verhindern, werden Änderungen in mehreren Testumgebungen getestet, damit es zu möglichst wenig Komplikationen auf dem produktiven Server kommt.

## Analyse des Bewilligungsverfahrens

Interne Mitarbeiter (Business) hat einen RFC zu entwerfen. Es können auch, in der Grafik nicht eingezeichnet, RFCs von den Entwicklern vorgeschlagen werden. Diese müssen jedoch durch das Business richtig eingereicht werden, deshalb ist es nicht eingezeichnet. Das Business gibt

den RFC an die Product Owner (PO, Projektleiter) weiter. Die entscheiden dann, ob ein RFC in die kommende Iteration (SAFE) passt, oder ob die Kapazität nicht ausreicht. Im zweiten Fall wird der RFC zurückgewiesen und zu einem späteren Zeitpunkt erneut geprüft. Passt der RFC in die Iteration, wird der RFC mit dem Entwicklungsteam besprochen. Empfindet das Entwicklungsteam den RFC als gefährlich, unnötig, nicht messbar oder der RFC ist für das Entwicklungsteam zu wenig Ausformuliert (Akzeptanzkriterien), muss der PO mit den Mängeln zum Business gehen und den RFC anpassen.

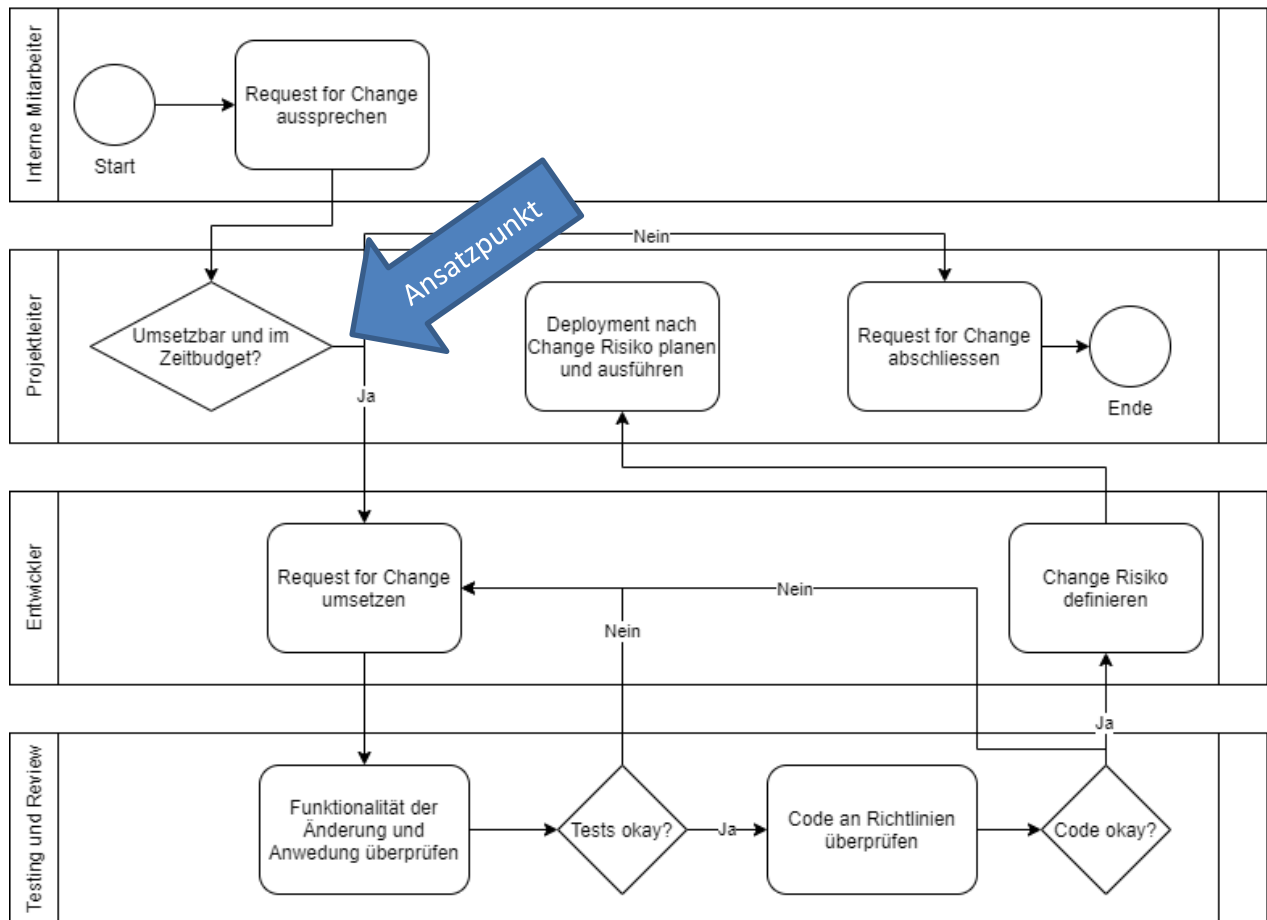
Das Entwicklungsteam kennt die Applikationen aus einer technischen Sicht und kann im Team besprechen, ob die Auswirkungen des RFCs weitere Änderungen mit sich ziehen würde. Hin und wieder stellt das Entwicklungsteam fest, dass Sicherheitsrisiken entstehen können, welche im RFC nicht berücksichtigt wurden, oder dass die Infrastruktur nicht zurechtgeschnitten ist: fehlende API's, Rechenleistung oder Konflikte mit anderen Applikationen. Auch in diesen Fällen muss der PO mit dem Business den RFC überarbeiten.

Anschliessend wird der beschriebene Prozess wiederholt.

Erfüllt ein RFC die Ansprüche des POs und des Entwicklungsteams wird der RFC umgesetzt. Somit ist der RFC bewilligt und das Bewilligungsverfahren ist abgeschlossen und kommt in die Umsetzung.

## Idealer

## Änderungsprozess



*RFC von Interdiscount mit neuem Ansatzpunkt.*

In der von mir vorgeschlagener Änderung des RFC-Prozesses will ich die Position des PO's stärken und diesem mehr Werkzeuge zur Verfügung stellen.

1. Der PO ist in direktem Kontakt mit dem Business und dem Entwicklungsteam.
2. Der PO besitzt eine gute Übersicht der Projekte.
3. Der PO weiss, wie sein Team funktioniert.

Diese Aspekte würde ich mehr ausnutzen, als es im Kapitel **Analyse des Bewilligungsverfahrens** beschrieben ist. Die Aktuellen Werkzeuge des POs in Bezug auf einen RFC:

- RFC aufgrund fehlender Kapazität in der Iteration verschieben.
- RFC aufgrund von Kritik des Teams an Business zurückgeben.

Beim zweiten Punkt setzt die vorgeschlagene Erweiterung an. Äussert das Team Kritik am RFC aufgrund nicht beachteter Risiken bei der Planung. Soll der PO der PO die Möglichkeit

erhalten, mit dem Team den RFC aufzuteilen in einen RFC, welcher keine absehbaren Risiken hat und in einen RFC, welcher am anderen RFC anknüpft, worin das Team planbare Risiken sieht. Also Risiken welche durch bessere Planungen minimiert werden können. Neu kann der PO folgende Werkzeuge einsetzen.

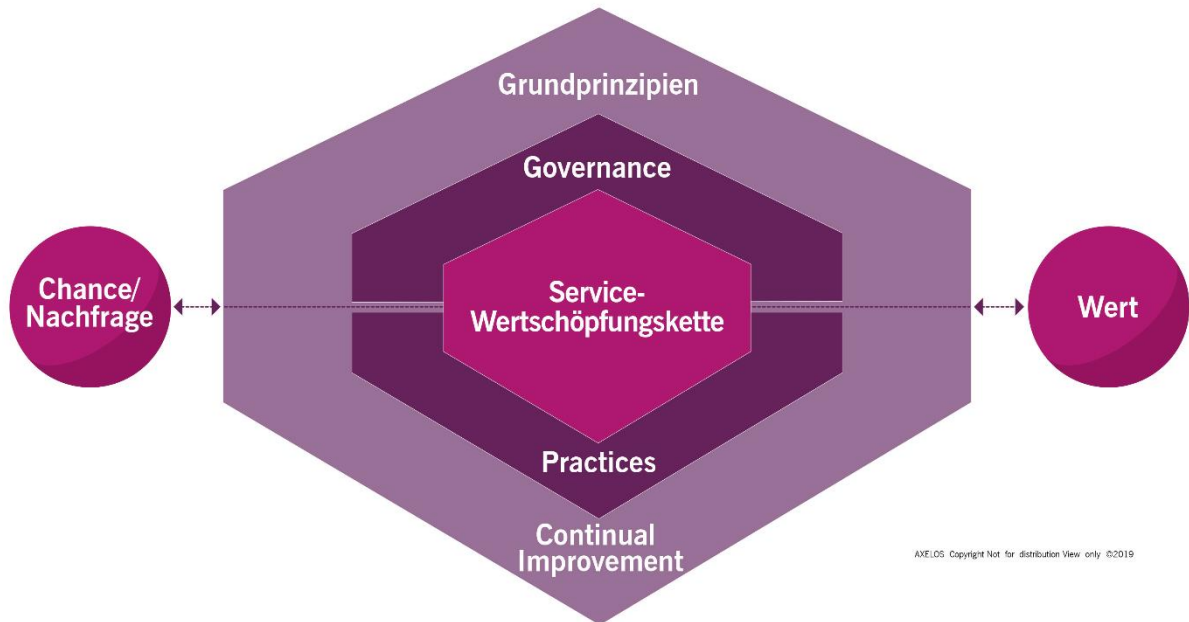
- RFC aufgrund fehlender Kapazität in der Iteration verschieben.
- RFC aufgrund von Kritik des Teams an Business zurückgeben.
- RFC aufteilen und zur neuen Aufbereitung an das Business zurückgeben.

Als Beispiel dient hier ein hypothetischer RFC bezüglich der Kontaktformular-Applikation. Das Entwicklungsteam ist vielleicht nicht damit einverstanden, wie der RFC in die Webapplikation eingebettet werden sollte. Den Rest des RFCs ist für das Team jedoch in Ordnung. Nun wird der RFC aufgeteilt in einen RFC, welche die Logik der Kontaktformular-Applikation beinhaltet und einen RFC, welcher die Einbettung der Applikation beinhaltet.

Diese Erweiterung verlangt mehr Planung. Die Vorteile überwiegen jedoch darin, dass ein RFC teilweise umgesetzt werden kann und somit die Kapazität einer Iteration weniger belastet wird.

Der Pilotbetrieb bleibt gleich wie vor der Erweiterung, erhält jedoch auch einen höheren Planungsaufwand, um die gesamte Applikation testen zu können.

## ITIL Service Value System



Die Komponenten von ITIL Service Value System müssen zusammenarbeiten. Es ist dafür da, um Flexibilität und gute Werte zu garantieren. Es ist universal und kann auf viele Bereiche, nicht nur in der Arbeitswelt, eingesetzt werden.

### **Opportunity/demand:**

Dies sind die Inputs im System. Es repräsentiert Optionen und Möglichkeiten von unseren Werten. Demand ist die Nachfrage eines Produkts. Es ist ein Mechanismus auf dem wir, nach Bedarf, handeln können, um unsere Werte aufrecht zu erhalten. Opportunity und demand bauen aufeinander auf, um das gemeinsame Ziel vom Value zu erreichen.

### **Guiding principles:**

Diese Prinzipie sind dauerhaft, universal und führen die Organisation durch den ganzen Prozess. Sie sind immer relevant und können immer verwendet werden. Sie können in der Arbeitswelt und im Privaten Leben eingesetzt werden. Es sind simple und vernünftige Regeln die öfters vergessen werden. Deswegen ist es wichtig diese zu artikulieren und zu dokumentieren. Sie zu einer Routine werden zu lassen damit wir sie automatisch anwenden. Es ist ein Mechanismus, um sicher zu stellen, dass das Richtige aufgebaut wird.

### **Governance:**

Governance ist dafür da die Regeln zu erstellen und diese Regeln und Prinzipien dann auch durchzusetzen. Es ist ein Kontrollmechanismus das dafür sorgt das Systeme in die richtige Richtung gesteuert werden. Jede Organisation besitzt die Governance in irgendeiner Form. Es

ist nicht notwendigerweise die Abteilung, welche die Arbeit erledigt, aber es ist zuständig für die erfolge oder Misserfolge. Die Verantwortung wird von der Governance getragen.

**Service value chain:**

Dies sind die Elemente im System. Es gibt mehrere Ketten die Operations-Model repräsentieren. Sie gliedern Gruppen von Komplizierte Aktivitäten miteinander. Das Ziel ist es den Input vom Demand zum Output zu bringen.

**Practices:**

Es beinhaltet Ressourcen die zur Verfügung gestellt werden, um Arbeiten zu erledigen. Ein Beispiel dafür ist der Servicedesk. Diese Prinzipien dienen dazu die Werte zu formen und sind ein wesentlicher Teil von diesem System.

**Continual improvement:**

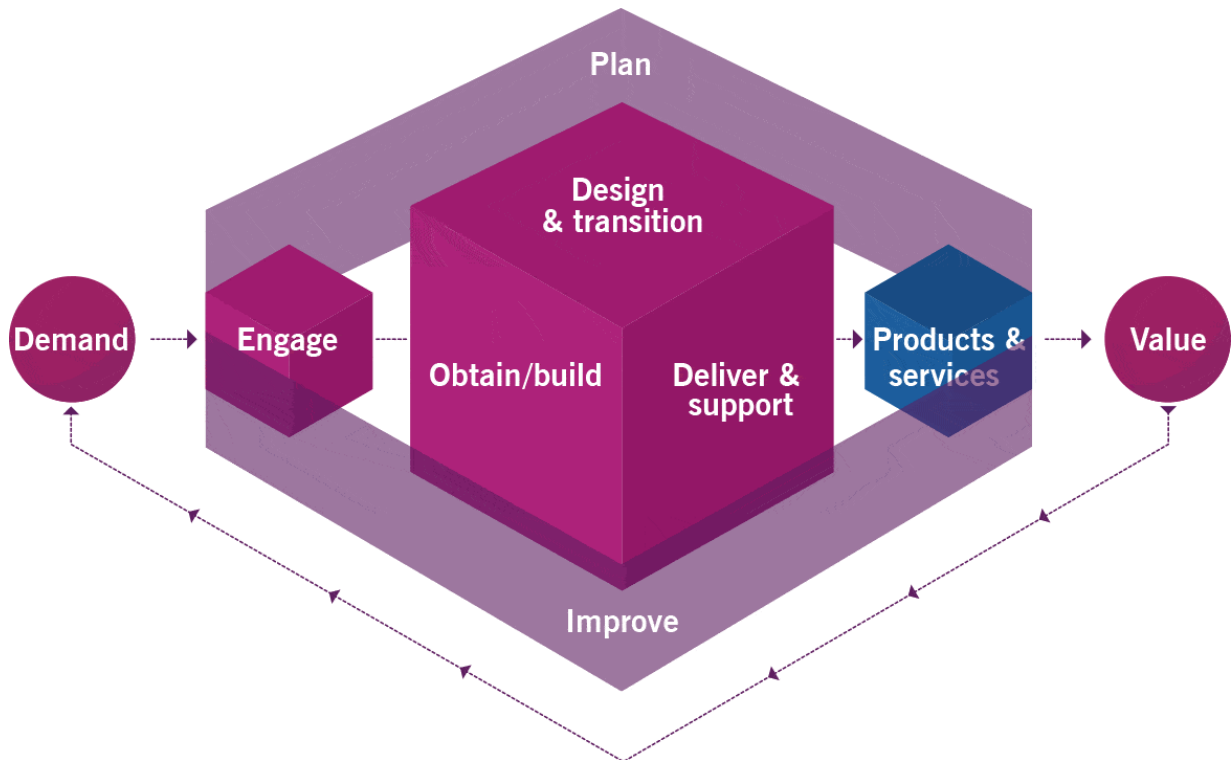
Jeder in der Organisation ist dafür zuständig Feedbacks zu geben damit das Unternehmen und Arbeitsprozesse optimiert werden können. Unterschiedliche Abteilungen und Firmen könne einander Feedbacks geben, um für die Kontinuierliche Verbesserung von Betrieben zu sorgen.

**Value:**

Dies ist der Output des Systems. Die Konsequenzen von den vorher gemachten Abänderungen. Es ist ein gemeinsam erschaffener Wert. Um den Value zu stande zu bringen, arbeiten Menschen und die verschiedenen Rollen zusammen, um den gewünschten Wert oder Value zu erhalten. Dies ist ein nie endender Prozess.

**ITIL Service Value Chain**





Für Service Provider ist die Wertschöpfungskette ein Betriebsmodell. Es beinhaltet alle wichtigen Aktivitäten, die für ein optimales Management der Produkte und Services wichtig sind. Jedes dieser Aktivitäten ist eine relevante Schnittstelle für die Stakeholder.

**Demand:**

Steht für Etwas, das repariert oder erstellt werden muss.

**Plan:**

Hier wird herausgefunden, wie die verbesserte Version aussehen sollte und die dafür notwendigen Schritte müssen geplant werden.

**Engage:**

Ein Team muss zusammengestellt werden und Ressourcen von aussen müssen an Bord gebracht werden.

**Design & Transition:**

Anschliessend muss herausgefunden werden, wie die Ressourcen verwendet werden und wie sie kombiniert werden. Eine Transition muss nach dem alles andere erledigt wurde stattfinden.

**Obtain/Build:**

In dieser Abteilung wird möglicher weisse zusätzliches wissen oder Ressourcen geholt.

**Deliver & Support:**

Hier wird das Produkt an den designierten Platz eigesetzt und supportet.

**Improve:**

Damit ist gemeint das Verbesserungen auf jedem Level stattfinden müssen.

## Präsentation idealer Änderungsprozess

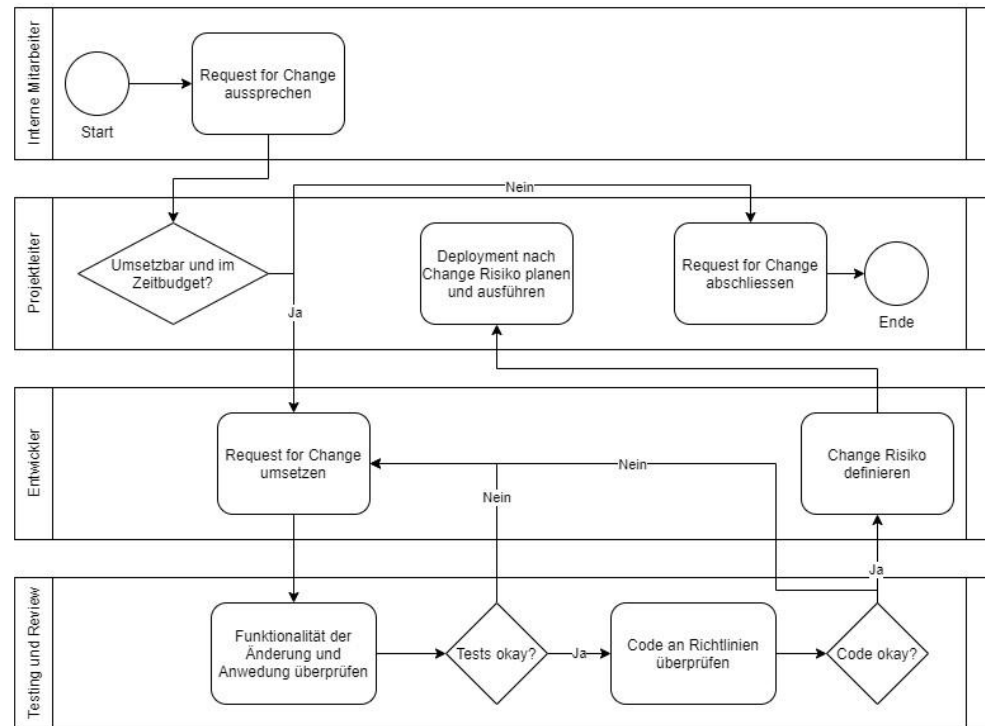


# Idealer Änderungsprozess

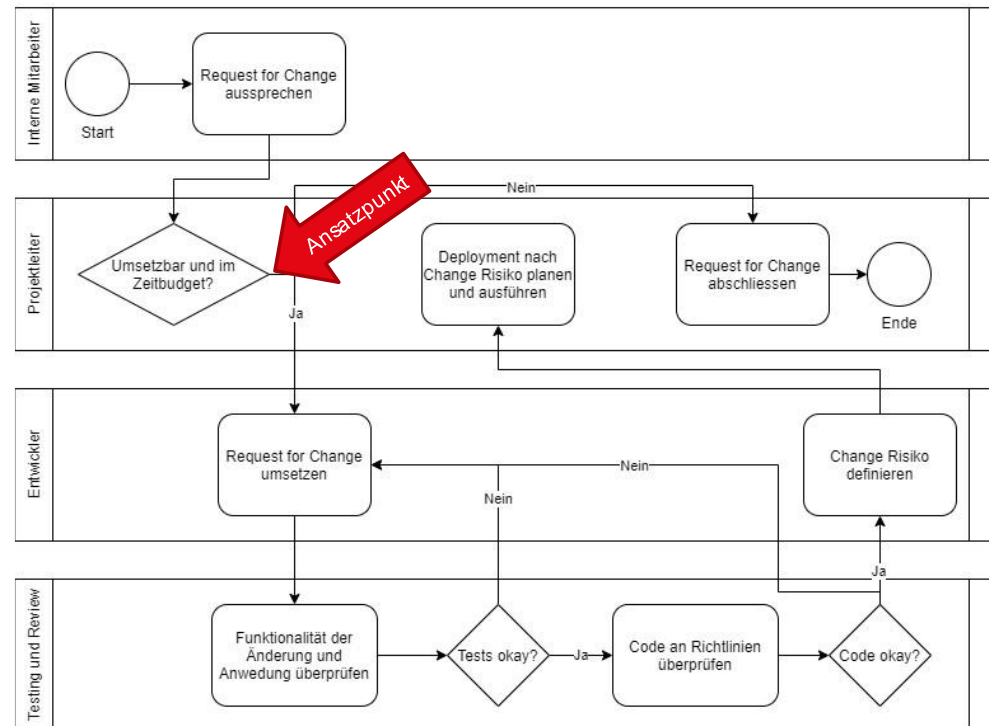
Modul 150

Sashauna Wray und Valentin Ehinger

## Aktueller Änderungsprozess



## Verbesserung des Prozesses



## Ansatzpunkt Product Owner (PO)

1. Der PO ist in direktem Kontakt mit dem Business und dem Entwicklungsteam.
2. Der PO besitzt eine gute Übersicht der Projekte.
3. Der PO weiss, wie sein Team funktioniert.

### Kompetenzen

- RFC aufgrund fehlender Kapazität in der Iteration verschieben.
- RFC aufgrund von Kritik des Teams an Business zurückgeben.

### Kompetenzen neu

- RFC aufgrund fehlender Kapazität in der Iteration verschieben.
- RFC aufgrund von Kritik des Teams an Business zurückgeben.
- RFC aufteilen und zur neuen Aufbereitung an das Business zurückgeben.

**Vielen Dank für Ihre Aufmerksamkeit!**

