

9.5 Zusammenspiel mit Incident, Change, Knowledge und Service Level Management

“ Kurz erklärt

Service Request Management arbeitet nicht isoliert.

Es steht in engem Zusammenhang mit anderen ITIL Practices.

Erst das Zusammenspiel mit Incident Management, Change Enablement, Knowledge Management, Service Level Management, Information Security Management und Continual Improvement ermöglicht eine schnelle, sichere und qualitativ hochwertige Bearbeitung von Service Requests.

Jede Practice liefert Informationen oder Leistungen, die den gesamten Request-Prozess unterstützen.

Warum das Zusammenspiel wichtig ist

Ein Service Request besteht selten nur aus einer einzelnen Tätigkeit.

Beispiel:

Ein neuer Mitarbeiter benötigt:

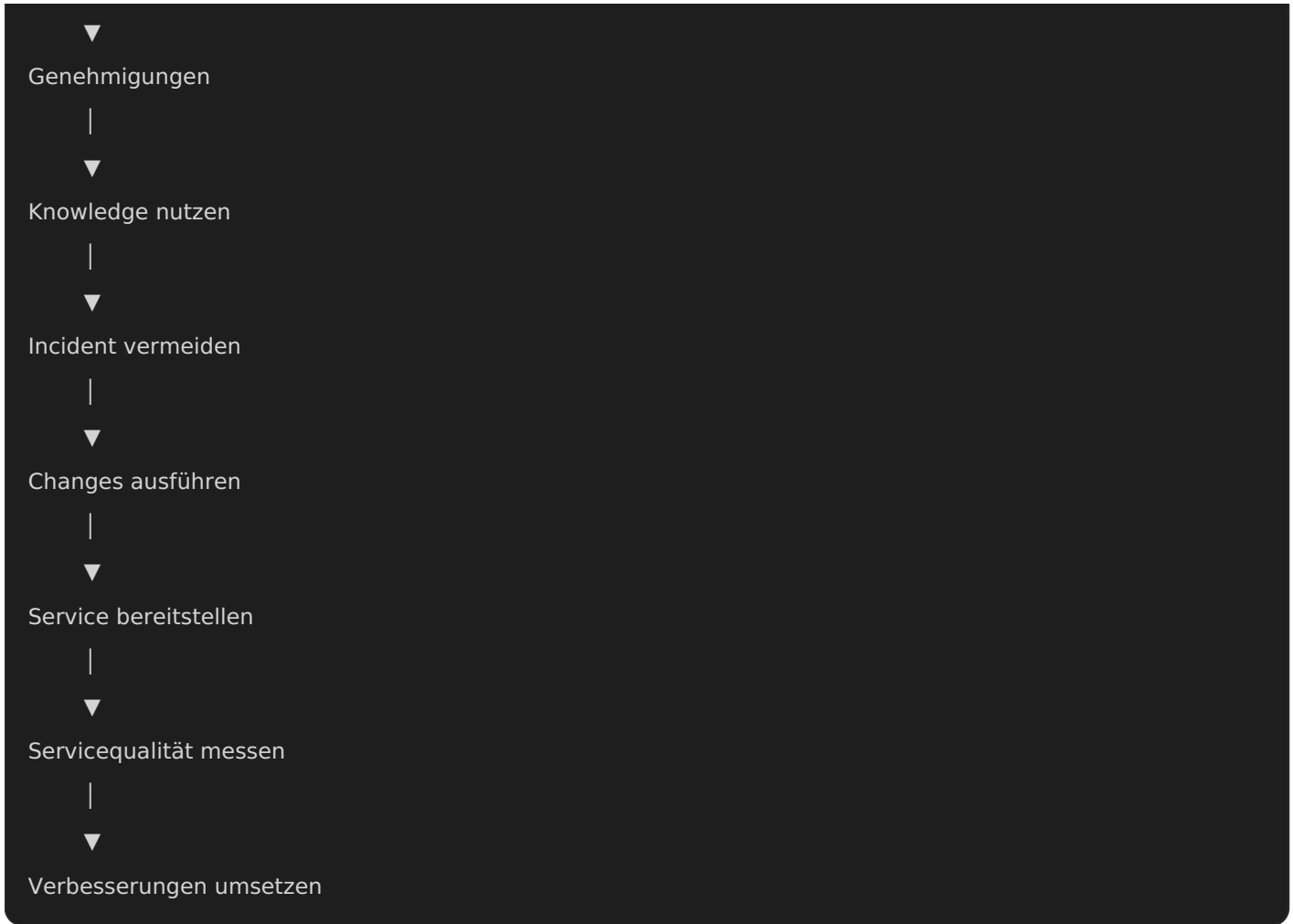
- Benutzerkonto,
- Microsoft-365-Lizenz,
- VPN,
- Notebook,
- Gruppenberechtigungen,
- Mobiltelefon.

Für die Bereitstellung arbeiten häufig mehrere Teams und Practices zusammen.

Das Service Request Management koordiniert dabei den gesamten Ablauf.

Zusammenspiel der Practices

Service Request



Die Practices arbeiten parallel und ergänzen sich gegenseitig.

Service Request Management und Incident Management

Diese beiden Practices werden häufig verwechselt.

Der wichtigste Unterschied:

Incident Management	Service Request Management
Störung beheben	Service bereitstellen
ungeplantes Ereignis	geplante Anfrage
Ziel: Wiederherstellung	Ziel: Erfüllung einer Anforderung
häufig hoher Zeitdruck	planbare Bearbeitung

Trotzdem arbeiten beide eng zusammen.

Wenn ein Incident zum Request wird

Manchmal stellt sich während der Bearbeitung heraus,

dass gar keine Störung vorliegt.

Beispiel:

Benutzer meldet:

“Ich kann Photoshop nicht benutzen.”

Analyse:

Photoshop wurde nie installiert.

Es liegt keine Störung vor.

Der Incident wird geschlossen.

Anschließend wird ein Service Request für die Softwareinstallation erstellt.

Wenn ein Request zum Incident wird

Auch das Gegenteil ist möglich.

Beispiel:

Ein Benutzer beantragt VPN.

Während der Bereitstellung fällt auf:

Der VPN-Gateway ist ausgefallen.

Jetzt liegt zusätzlich ein Incident vor.

Der Service Request bleibt bestehen,

die technische Störung wird jedoch separat als Incident bearbeitet.

Parallele Bearbeitung

Ein Service Request und ein Incident können gleichzeitig existieren.

Beispiel:

- Request: neues Notebook
- Incident: Softwareverteilung funktioniert nicht

Beide Vorgänge besitzen unterschiedliche Ziele,
können sich jedoch gegenseitig beeinflussen.

Verknüpfung von Tickets

Viele ITSM-Systeme ermöglichen Verknüpfungen.

Beispiel:



Dadurch bleiben Zusammenhänge nachvollziehbar.

Service Request Management und Change Enablement

Viele Requests führen zu Änderungen an der IT-Umgebung.

Beispiele:

- Software installieren,
- Benutzerkonto erstellen,
- Firewallregel hinzufügen,
- virtuelle Maschine bereitstellen,
- Cloud-Service aktivieren.

Nicht jede dieser Änderungen benötigt jedoch einen eigenen Change.

Standard Changes

Viele Service Requests werden über bereits genehmigte Standard Changes umgesetzt.

Beispiele:

- Office installieren

- Gruppenmitgliedschaft vergeben
- Notebook bereitstellen
- Benutzerkonto anlegen

Der Benutzer bemerkt davon häufig nichts.

Für ihn bleibt es ein Service Request.

Normale Changes

Größere Anforderungen können zusätzlich einen normalen Change benötigen.

Beispiel:

Ein Fachbereich benötigt eine neue Unternehmenssoftware.

Der Service Request startet den Prozess.

Die technische Einführung erfolgt anschließend kontrolliert über Change Enablement.

Service Request Management und Knowledge Management

Knowledge Management unterstützt Service Requests in nahezu jeder Phase.

Beispiele:

- Benutzer findet die Lösung selbst.
- Service Desk nutzt einen Knowledge-Artikel.
- Automatisierung greift auf Dokumentation zurück.
- Runbooks beschreiben die Bereitstellung.

Je besser die Wissensbasis,

desto schneller erfolgt die Bearbeitung.

Knowledge vor Ticket

Idealerweise prüft der Benutzer bereits vor der Ticketerstellung,

ob ein passender Knowledge-Artikel existiert.

Beispiele:

- VPN einrichten
- Outlook konfigurieren

- Drucker verbinden
- MFA aktivieren

Dadurch werden viele Requests vollständig vermieden.

Neue Knowledge-Artikel

Auch Service Requests können neue Dokumentationen erzeugen.

Beispiele:

- häufige Rückfragen
- neue Software
- neue Cloud-Dienste
- neue Standardhardware
- neuer Bereitstellungsprozess

Wissen sollte kontinuierlich erweitert werden.

Runbooks

Viele Standard Requests besitzen ein Runbook.

Ein Runbook beschreibt:

- Voraussetzungen,
- Reihenfolge,
- Prüfungen,
- Dokumentation,
- Abschluss.

Dadurch arbeiten alle Bearbeiter nach demselben Verfahren.

Service Request Management und Service Level Management

Auch Service Requests besitzen häufig Serviceziele.

Beispiele:

- Notebook innerhalb von fünf Arbeitstagen
- Benutzerkonto innerhalb von vier Stunden
- Passwort innerhalb von zehn Minuten
- Software innerhalb eines Arbeitstages

Service Level Management überwacht,

ob diese Ziele erreicht werden.

Kennzahlen

Typische Kennzahlen:

- Bearbeitungszeit
- Wartezeit auf Genehmigung
- Automatisierungsquote
- Anzahl offener Requests
- durchschnittliche Bereitstellungszeit
- Benutzerzufriedenheit
- Self-Service-Nutzung
- Wiedereröffnungen

Diese Kennzahlen helfen bei Verbesserungen.

Service Request Management und Information Security Management

Viele Requests betreffen Sicherheitsaspekte.

Beispiele:

- Administratorrechte
- VPN
- Multifaktor-Authentifizierung
- Cloud-Zugriffe
- Gruppenmitgliedschaften
- Datenbankberechtigungen

Information Security Management definiert häufig die notwendigen Anforderungen.

Least Privilege

Service Requests sollten möglichst nach dem Prinzip der geringsten Berechtigung umgesetzt werden.

Benutzer erhalten nur die Rechte,

die sie tatsächlich benötigen.

Dadurch sinkt das Sicherheitsrisiko.

Rezertifizierung

Manche Berechtigungen werden regelmäßig überprüft.

Beispiele:

- Administratorrechte
- Projektzugriffe
- externe Benutzer
- Cloud-Rollen

Nicht mehr benötigte Berechtigungen werden entfernt.

Service Request Management und Service Configuration Management

Viele Requests verändern Configuration Items.

Beispiele:

- Notebook ausgeben
- Server bereitstellen
- virtuelle Maschine erzeugen
- Software installieren
- Zertifikat erneuern

Nach der Bereitstellung sollten die entsprechenden Configuration-Daten aktualisiert werden.

Warum Configuration-Daten wichtig sind

Nur aktuelle Daten ermöglichen:

- korrekte Inventarisierung,
- Incident-Analyse,
- Problem Management,
- Change Enablement,
- Lizenzverwaltung,
- Sicherheitsprüfungen.

Service Requests verändern häufig genau diese Informationen.

Service Request Management und Supplier Management

Externe Lieferanten können an der Bearbeitung beteiligt sein.

Beispiele:

- Notebooklieferung

- Mobiltelefon
- Cloud-Service
- Softwarelizenz
- Hardwarewartung

Lieferzeiten externer Partner beeinflussen häufig die gesamte Bearbeitungsdauer.

Service Request Management und Continual Improvement

Jeder Request liefert Verbesserungspotenzial.

Beispiele:

- unnötige Genehmigungen,
- schlechte Formulare,
- fehlende Automatisierung,
- veraltete Knowledge-Artikel,
- lange Bearbeitungszeiten,
- wiederkehrende Rückfragen.

Diese Erkenntnisse fließen in Continual Improvement ein.

Verbesserungsmöglichkeiten

Typische Maßnahmen:

- Formular vereinfachen
- Genehmigungen reduzieren
- Workflow automatisieren
- Knowledge erweitern
- Serviceportal verbessern
- Bearbeitungszeiten verkürzen
- Kommunikation verbessern

Kleine Verbesserungen können große Wirkung entfalten.

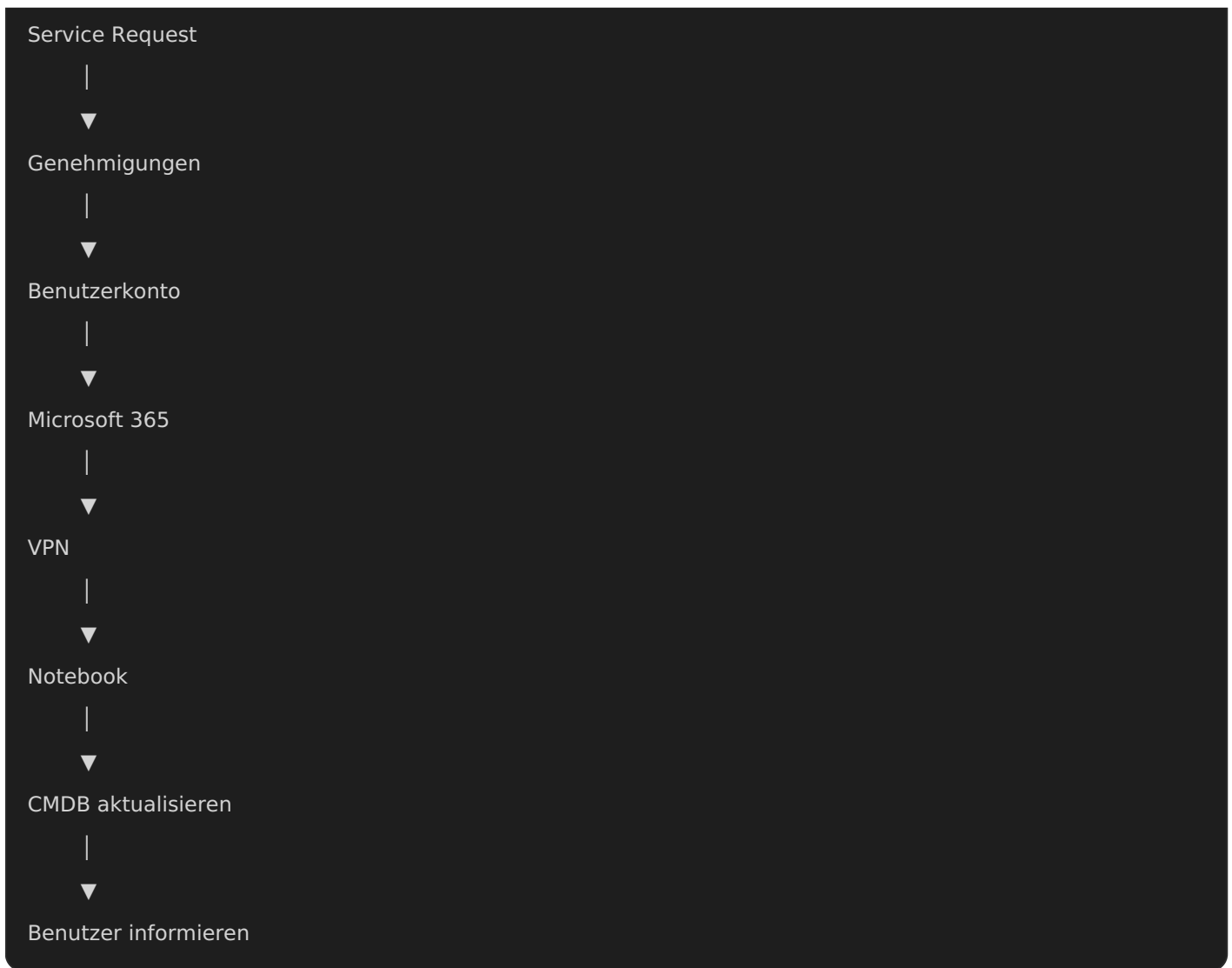
Zusammenspiel im Onboarding

Ein Beispiel:

Neuer Mitarbeiter

|





Mehrere Practices arbeiten dabei gleichzeitig zusammen.

Praxisbeispiel

Ein neuer Mitarbeiter beginnt nächste Woche.

Der Request löst automatisch aus:

- Benutzerkonto,
- E-Mail,
- Microsoft 365,
- Notebook,
- VPN,
- Gruppen,
- Teams,
- Mobiltelefon.

Währenddessen:

- Knowledge liefert Installationsanleitungen.
- Configuration Management dokumentiert neue Geräte.
- Service Level Management misst die Bearbeitungszeit.
- Information Security prüft Berechtigungen.
- Continual Improvement sammelt Kennzahlen.

Der Benutzer erlebt einen einzigen zusammenhängenden Service.

Typische Fehler

Fehler 1

Incidents und Requests werden verwechselt.

Fehler 2

Service Requests aktualisieren keine Configuration-Daten.

Fehler 3

Knowledge wird nicht genutzt.

Fehler 4

Genehmigungen verzögern Standardleistungen.

Fehler 5

Automatisierungen fehlen.

Fehler 6

Serviceziele werden nicht gemessen.

Fehler 7

Information Security wird zu spät eingebunden.

Fehler 8

Bearbeitung erfolgt außerhalb des ITSM-Systems.

Fehler 9

Self-Service wird kaum genutzt.

Fehler 10

Verbesserungspotenziale werden nicht ausgewertet.

Checkliste Zusammenspiel

- ☐ Request korrekt klassifiziert
 - ☐ Incident ausgeschlossen
 - ☐ Knowledge genutzt
 - ☐ Genehmigungen durchgeführt
 - ☐ Change erforderlich?
 - ☐ Configuration-Daten aktualisiert
 - ☐ Sicherheitsanforderungen erfüllt
 - ☐ Serviceziel überwacht
 - ☐ Benutzer informiert
 - ☐ Verbesserungen dokumentiert
-

Checkliste Automatisierung

- ☐ Workflow vorhanden
 - ☐ Genehmigungen integriert
 - ☐ Knowledge eingebunden
 - ☐ Configuration aktualisiert
 - ☐ Benutzer informiert
 - ☐ Fehler protokolliert
 - ☐ Monitoring vorhanden
 - ☐ Dokumentation aktuell
-

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker arbeiten täglich an den Schnittstellen verschiedener ITIL Practices.

Im Arbeitsalltag bedeutet dies:

- Service Requests technisch umsetzen,
- Incidents von Requests unterscheiden,
- Standard Changes anwenden,
- Configuration-Daten pflegen,
- Knowledge erweitern,
- Sicherheitsvorgaben einhalten,
- Automatisierungen entwickeln,
- Prozesse kontinuierlich verbessern.

Erst das Zusammenspiel aller Practices ermöglicht einen professionellen und effizienten IT-Service.

Zusammenfassung

“ Benutzer stellt Service Request

↓

Knowledge und Self-Service unterstützen

↓

Genehmigungen prüfen

↓

Standard Change nutzen (falls erforderlich)

↓

Service bereitstellen

↓

Configuration aktualisieren

↓

Servicequalität messen

↓

Verbesserungen umsetzen

Merksätze

Service Request Management arbeitet niemals isoliert.

“ Incidents beheben Störungen – Service Requests stellen Leistungen bereit.

“ Knowledge und Automatisierung beschleunigen die Bearbeitung.

“ Configuration-Daten müssen nach der Bereitstellung aktuell sein.

“ Continual Improvement verbessert auch Service Requests kontinuierlich.

Verwandte Seiten

- 9.1 Ziele und Grundlagen des Service Request Management
- 9.2 Request Model und Standard Requests
- 9.3 Genehmigungen und Freigabeverfahren
- 9.4 Erfüllung, Automatisierung und Self-Service
- Incident Management
- Change Enablement
- Knowledge Management
- Service Configuration Management
- Service Level Management
- Information Security Management
- Continual Improvement

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Request Management
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- PeopleCert – ITIL Practice Guide: Knowledge Management
- PeopleCert – ITIL Practice Guide: Service Configuration Management
- PeopleCert – ITIL Practice Guide: Service Level Management
- PeopleCert – ITIL Practice Guide: Information Security Management

- PeopleCert – ITIL Practice Guide: Continual Improvement
- ITIL Foundation – Version 5

Einordnung

Die dargestellten Zusammenhänge, Beispiele und Praxisempfehlungen orientieren sich an den ITIL-Practices. Die konkrete technische Umsetzung hängt von den eingesetzten ITSM-Systemen, Automatisierungsplattformen, Identity-Management-Lösungen und organisatorischen Prozessen der jeweiligen Organisation ab.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
