

3.8 Self-Service, Wissensnutzung und Automatisierung (Teil 2/2)

Runbooks und Standardabläufe

Runbooks beschreiben wiederholbare Arbeitsabläufe.

Sie helfen dabei, Aufgaben einheitlich, sicher und nachvollziehbar durchzuführen.

Beispiele:

- Server kontrolliert neu starten
- Backup-Wiederherstellung prüfen
- Zertifikat erneuern
- Benutzerkonto sperren
- DNS-Auflösung prüfen
- Dienststatus kontrollieren
- Speicherplatz erweitern
- Logdateien auswerten

Ein gutes Runbook enthält:

- Zweck des Ablaufs
- Voraussetzungen
- benötigte Berechtigungen
- einzelne Arbeitsschritte
- Prüfpunkte
- Risiken
- Rollback oder Abbruchkriterien
- Dokumentationshinweise

Runbook und Knowledge-Artikel unterscheiden

Runbook	Knowledge-Artikel
richtet sich häufig an IT-Mitarbeiter	richtet sich je nach Inhalt an Benutzer oder IT
beschreibt interne Arbeitsabläufe	beschreibt Lösungen, Erklärungen oder Anleitungen
enthält technische Prüfschritte	kann auch einfache Benutzerhilfe enthalten
benötigt oft Berechtigungen	kann öffentlich im Serviceportal stehen
dient der Standardisierung	dient der Wiederverwendung von Wissen

Beide Formen unterstützen eine schnellere und einheitlichere Bearbeitung.

Standardisierung

Standardisierung bedeutet, wiederkehrende Aufgaben nach einem einheitlichen Muster auszuführen.

Beispiele:

- einheitlicher Onboarding-Prozess
- definierte Softwarepakete
- standardisierte Berechtigungsgruppen
- festgelegte Namenskonventionen
- einheitliche Ticketkategorien
- wiederverwendbare Kommunikationsvorlagen
- geprüfte Standard-Changes
- dokumentierte Wiederherstellungsverfahren

Standardisierung reduziert:

- Fehler,
 - Rückfragen,
 - Sonderfälle,
 - Bearbeitungszeit,
 - und Abhängigkeit von Einzelpersonen.
-

Standardisierung ist keine Starrheit

Standardisierung bedeutet nicht, dass jede Situation gleich behandelt werden muss.

Sie schafft einen sicheren Normalweg.

Abweichungen bleiben möglich, sollten aber bewusst entschieden und dokumentiert werden.

Beispiel:

Ein Standard-Request für Softwareinstallation kann automatisch laufen.

Eine sicherheitskritische Spezialsoftware benötigt dagegen zusätzliche Prüfung und Freigabe.

Automatisierung sinnvoll einsetzen

Automatisierung sollte dort eingesetzt werden, wo Aufgaben:

- häufig wiederkehren,
- klar beschrieben sind,
- wenig Interpretationsspielraum besitzen,
- sicher überprüfbar sind,

- und bei Fehlern kontrolliert zurückgenommen werden können.

Gute Kandidaten:

- Passwort-Reset
- Benutzeranlage
- Gruppenmitgliedschaften
- Softwarebereitstellung
- Ticket-Routing
- Statusbenachrichtigungen
- Standard-Reports
- Zertifikatsprüfung
- Speicherplatzwarnungen

Weniger geeignet sind Aufgaben mit:

- hoher Unsicherheit,
- unklarer Datenlage,
- hohem Sicherheitsrisiko,
- vielen Ausnahmen,
- oder notwendiger Einzelfallentscheidung.

Automatisierung braucht Kontrolle

Eine Automatisierung ist nur dann hilfreich, wenn sie zuverlässig gesteuert wird.

Zu klären ist:

- Wer besitzt die Automatisierung?
- Wer darf sie ändern?
- Wie wird sie getestet?
- Wie werden Fehler erkannt?
- Wie erfolgt ein Rollback?
- Welche Logs werden geschrieben?
- Welche Berechtigungen nutzt sie?
- Welche Daten werden verarbeitet?
- Wie wird Missbrauch verhindert?

“ Merke

Automatisierung entfernt Arbeit nicht vollständig. Sie verlagert Arbeit in Design, Prüfung, Betrieb und Überwachung.

Risiken fehlerhafter Automatisierung

Fehlerhafte Automatisierungen können große Auswirkungen haben.

Beispiele:

- falsche Benutzer werden angelegt
- falsche Berechtigungen werden vergeben
- Konten werden versehentlich deaktiviert
- Software wird auf falschen Geräten installiert
- Daten werden überschrieben
- Tickets werden falsch priorisiert
- Benachrichtigungen werden nicht versendet
- Zertifikate werden falsch erneuert

Deshalb benötigen Automatisierungen:

- Tests,
- Freigaben,
- Protokollierung,
- Monitoring,
- Fehlerbehandlung,
- und klare Verantwortlichkeiten.

Self-Service und Automatisierung kombinieren

Self-Service wird besonders wirksam, wenn Benutzeranfragen im Hintergrund automatisiert bearbeitet werden.

Beispiel:

Ein Benutzer beantragt Standardsoftware im Serviceportal.

Möglicher Ablauf:

Benutzer wählt Software im Portal



Berechtigung und Genehmigung werden geprüft



Softwarepaket wird automatisch zugewiesen



Installation startet auf dem Gerät



Benutzer erhält Statusmeldung



Ticket wird automatisch dokumentiert

Dadurch muss der Service Desk nicht jeden Standardschritt manuell durchführen.

Genehmigungen einbauen

Nicht jede Anfrage darf sofort automatisiert erfüllt werden.

Beispiele für genehmigungspflichtige Requests:

- kostenpflichtige Software
- erhöhte Berechtigungen
- Zugriff auf vertrauliche Daten
- externe Freigaben
- neue Geräte
- Administratorrechte
- Cloud-Ressourcen mit Kostenfolge

Ein guter Workflow unterscheidet zwischen:

- automatisch erfüllbaren Standardanfragen,
 - Anfragen mit Genehmigung,
 - und Anfragen mit manueller Prüfung.
-

Self-Service muss benutzerfreundlich sein

Ein Self-Service-Portal bringt wenig, wenn Benutzer es nicht verstehen oder nicht finden.

Wichtig sind:

- klare Sprache,
- gute Suche,
- verständliche Kategorien,
- einfache Formulare,
- kurze Beschreibungen,
- sichtbarer Bearbeitungsstatus,
- mobile Nutzbarkeit,
- und sinnvolle Hilfetexte.

Benutzer sollten nicht wissen müssen, welches interne Team zuständig ist.

Ungeeignet:



Antrag für AD-Gruppenmitgliedschaft CN=APP-FIN-PRD-RW.

Besser:

“ Zugriff auf Finanzanwendung beantragen.

Gute Formulare

Ein Formular sollte genau die Informationen erfassen, die für die Bearbeitung notwendig sind.

Beispiele:

- betroffener Service
- gewünschte Leistung
- betroffene Person
- gewünschter Zeitpunkt
- Begründung
- Kostenstelle
- benötigte Berechtigung
- Genehmiger
- betroffene Geräte
- Standort

Zu viele Pflichtfelder führen dazu, dass Benutzer falsche Angaben machen oder den Service Desk direkt kontaktieren.

Zu wenige Pflichtfelder führen zu Rückfragen und Verzögerungen.

Status transparent machen

Benutzer sollten erkennen können:

- ob die Anfrage eingegangen ist,
- wer aktuell zuständig ist,
- ob eine Genehmigung fehlt,
- welcher nächste Schritt erfolgt,
- wann eine Rückmeldung zu erwarten ist,
- und ob ein Workaround verfügbar ist.

Transparenz reduziert Rückfragen.

Beispiel:

Ihre Softwareanfrage wartet aktuell auf Genehmigung durch den Fachbereich.
Nach Genehmigung startet die automatische Bereitstellung.

Wissensartikel aus Tickets erzeugen

Viele gute Knowledge-Artikel entstehen direkt aus realen Tickets.

Nach einer Lösung sollte geprüft werden:

- War der Fall wiederholbar?
- Wird diese Frage wahrscheinlich erneut auftreten?
- Ist die Lösung allgemein nutzbar?
- Kann der Service Desk sie künftig selbst anwenden?
- Kann ein Benutzer sie selbst durchführen?
- Muss ein bestehender Artikel angepasst werden?

Nicht jeder Incident benötigt einen neuen Artikel.

Aber häufige oder lehrreiche Fälle sollten nicht verloren gehen.

Qualität von Knowledge-Artikeln sichern

Knowledge-Artikel sollten regelmäßig geprüft werden.

Zu prüfen ist:

- Ist der Inhalt noch korrekt?
- Stimmen Screenshots und Menüpfade?
- Gibt es neue Versionen?
- Sind Links noch gültig?
- Sind Risiken vollständig beschrieben?
- Ist die Zielgruppe passend?
- Gibt es Rückmeldungen von Benutzern?
- Wird der Artikel tatsächlich gefunden?
- Führt der Artikel zu weniger Tickets?

Veraltetes Wissen kann schädlicher sein als kein Wissen.

Feedback nutzen

Benutzer und Service-Desk-Mitarbeiter sollten Feedback geben können.

Beispiele:

- Artikel hilfreich oder nicht hilfreich
- Suchbegriff führte nicht zum Ergebnis
- Schritt unverständlich
- Screenshot veraltet
- Lösung funktioniert nicht mehr
- Formular enthält unklare Felder
- Automatisierung hat falschen Status geliefert

Dieses Feedback sollte regelmäßig ausgewertet werden.

Kennzahlen

Mögliche Kennzahlen für Self-Service, Wissensnutzung und Automatisierung:

Bereich	Mögliche Kennzahl
Self-Service	Anteil der Anfragen über Portal
Knowledge Base	Artikelaufrufe
Knowledge Base	hilfreiche Bewertungen
Service Desk	First Contact Resolution
Automatisierung	automatisch erfüllte Requests
Automatisierung	Fehlerquote automatisierter Abläufe
Service Requests	durchschnittliche Erfüllungszeit
Benutzererfahrung	Zufriedenheit nach Nutzung
Qualität	Anzahl veralteter Artikel
Verbesserung	reduzierte Wiederholungstickets

Kennzahlen sollten nicht nur Menge messen, sondern auch Qualität.

Problematische Kennzahlen

Nicht jede Kennzahl führt automatisch zu gutem Verhalten.

Beispiel:

Viele Knowledge-Artikel

Kann problematisch sein, wenn viele Artikel veraltet, doppelt oder unverständlich sind.

Besser:

- hilfreiche Artikel,

- aktuelle Artikel,
- gute Suchtreffer,
- weniger Rückfragen.

Beispiel:

Viele automatisierte Tickets

Kann problematisch sein, wenn Automatisierungen Fehler erzeugen oder Benutzer nicht verstehen, was passiert.

Besser:

- erfolgreiche Automatisierung,
- geringe Fehlerquote,
- nachvollziehbare Ergebnisse,
- klare Rückmeldung an Benutzer.

Zusammenspiel mit Incident Management

Self-Service und Knowledge Management unterstützen Incident Management.

Beispiele:

- Benutzer lösen einfache Störungen selbst.
- Service Desk findet schneller bekannte Lösungen.
- Workarounds werden einheitlich kommuniziert.
- Wiederkehrende Incidents werden sichtbar.
- Known Errors können dokumentiert werden.
- Automatisierte Diagnosen liefern erste Hinweise.

Dadurch kann der Service schneller wiederhergestellt werden.

Zusammenspiel mit Service Request Management

Viele Service Requests eignen sich besonders gut für Self-Service und Automatisierung.

Beispiele:

- Standardsoftware anfordern
- Zugriff beantragen
- Hardware bestellen
- Benutzerkonto erstellen
- Verteilerliste anpassen
- Passwort zurücksetzen

Wichtig ist, dass:

- der Service verständlich beschrieben ist,
 - Genehmigungen klar sind,
 - Bearbeitungszeiten sichtbar sind,
 - und die Erfüllung nachvollziehbar dokumentiert wird.
-

Zusammenspiel mit Change Enablement

Automatisierung und Self-Service können Changes auslösen.

Beispiele:

- neue Berechtigung wird gesetzt
- Software wird installiert
- Firewall-Regel wird beantragt
- Cloud-Ressource wird erstellt
- Konfiguration wird angepasst

Solche Änderungen müssen entsprechend ihrem Risiko behandelt werden.

Nicht jede Automatisierung ist automatisch ein Standard-Change.

Die Organisation muss festlegen, welche Abläufe vorab geprüft und freigegeben sind.

Zusammenspiel mit Information Security Management

Self-Service und Automatisierung betreffen häufig Berechtigungen und Daten.

Deshalb müssen Sicherheitsanforderungen berücksichtigt werden.

Beispiele:

- starke Authentifizierung
- Rollen- und Rechteprüfung
- Genehmigungsworkflow
- Protokollierung
- Zugriff nur nach Bedarf
- regelmäßige Rezertifizierung
- Schutz sensibler Daten
- Missbrauchserkennung

Benutzerfreundlichkeit darf nicht dazu führen, dass Sicherheitsregeln umgangen werden.

Typische Fehler

Fehler 1

Self-Service wird nur als Ticketformular verstanden.

Fehler 2

Benutzer müssen interne IT-Strukturen kennen.

Fehler 3

Knowledge-Artikel sind veraltet.

Fehler 4

Artikel sind zu technisch für Endbenutzer.

Fehler 5

Automatisierungen werden ohne ausreichende Tests produktiv genutzt.

Fehler 6

Automatisierungen besitzen zu viele Berechtigungen.

Fehler 7

Fehler in automatisierten Abläufen werden nicht überwacht.

Fehler 8

Runbooks existieren, werden aber nicht gepflegt.

Fehler 9

Self-Service ersetzt persönliche Hilfe auch dort, wo sie notwendig wäre.

Fehler 10

Benutzerfeedback wird nicht ausgewertet.

Fehler 11

Genehmigungen sind unklar oder zu langsam.

Fehler 12

Wissen bleibt bei Einzelpersonen statt in dokumentierten Systemen.

Praxisbeispiel: Passwort-Reset

Ausgangslage

Viele Benutzer rufen den Service Desk an, weil sie ihr Passwort vergessen haben.

Verbesserung

Ein Self-Service-Passwort-Reset wird eingeführt.

Wichtige Anforderungen

- sichere Identitätsprüfung
- MFA-Unterstützung
- klare Benutzeranleitung
- Protokollierung
- Sperrmechanismen gegen Missbrauch
- Hilfeweg bei Fehlern

Nutzen

- weniger Standardkontakte
 - schnellere Hilfe für Benutzer
 - Service Desk hat mehr Zeit für komplexe Incidents
-

Praxisbeispiel: Softwareanforderung

Ausgangslage

Software wird per E-Mail angefordert.

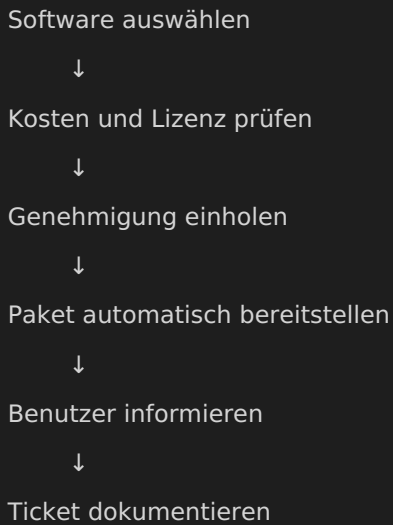
Informationen fehlen häufig.

Genehmigungen sind unklar.

Verbesserung

Ein Serviceportal bietet standardisierte Softwareanfragen.

Ablauf



Nutzen

- weniger Rückfragen
- klare Genehmigung
- schnellere Bereitstellung
- bessere Nachvollziehbarkeit

Praxisbeispiel: Knowledge-Artikel aus Incident

Incident

Mehrere Benutzer melden, dass VPN nach einem Client-Update nicht startet.

Lösung

Der Service Desk findet eine sichere Reparaturmaßnahme.

Folgeaktivität

Ein interner Knowledge-Artikel wird erstellt.

Inhalt:

- betroffene Version
- Symptom
- Prüfschritte
- Lösung
- Abbruchkriterien

- Eskalation an Netzwerkteam

Nutzen

Der nächste gleichartige Incident kann schneller gelöst werden.

Checkliste Self-Service

- ☐ Leistungen sind verständlich beschrieben
 - ☐ Benutzer müssen keine internen Teamnamen kennen
 - ☐ Formulare erfassen die richtigen Informationen
 - ☐ Status ist für Benutzer sichtbar
 - ☐ Genehmigungen sind klar geregelt
 - ☐ Hilfeweg bei Problemen ist vorhanden
 - ☐ Portal ist leicht auffindbar
 - ☐ Inhalte werden regelmäßig verbessert
-

Checkliste Knowledge Management

- ☐ Artikel besitzen klare Titel
 - ☐ Zielgruppe ist definiert
 - ☐ Schritte sind verständlich
 - ☐ Voraussetzungen sind genannt
 - ☐ Risiken sind beschrieben
 - ☐ Inhalte sind aktuell
 - ☐ Artikel werden gefunden
 - ☐ Feedback wird ausgewertet
 - ☐ veraltete Artikel werden überarbeitet oder entfernt
 - ☐ Lösungen aus Tickets werden bei Bedarf übernommen
-

Checkliste Automatisierung

- ☐ Ablauf ist standardisiert
- ☐ Risiken sind bewertet
- ☐ Berechtigungen sind minimal notwendig
- ☐ Tests wurden durchgeführt
- ☐ Fehler werden protokolliert

- ☐ Monitoring ist vorhanden
 - ☐ Rollback oder Abbruchweg ist definiert
 - ☐ Verantwortlicher ist benannt
 - ☐ Änderungen sind dokumentiert
 - ☐ Benutzer erhalten verständliche Rückmeldungen
-

Bedeutung für Fachinformatiker für Systemintegration

Für Fachinformatiker ist dieses Thema besonders wichtig, weil viele praktische Verbesserungen direkt aus dem technischen Alltag entstehen.

Beispiele:

- wiederkehrende Fehler dokumentieren,
- Standardabläufe als Runbook beschreiben,
- einfache Prüfungen automatisieren,
- Service Desk mit Wissen unterstützen,
- sichere Self-Service-Prozesse vorbereiten,
- Berechtigungen sauber modellieren,
- und Monitoring für Automatisierungen einrichten.

Gute technische Arbeit zeigt sich nicht nur darin, ein einzelnes Problem zu lösen.

Sie zeigt sich auch darin, dass derselbe Fehler künftig schneller, sicherer oder automatisch bearbeitet werden kann.

Zusammenfassung

“ Wiederkehrende Anfragen erkennen
↓
geeignete Self-Service-Möglichkeiten schaffen
↓
Wissen verständlich dokumentieren
↓
Standardabläufe definieren
↓
geeignete Schritte automatisieren
↓
Risiken, Berechtigungen und Monitoring berücksichtigen
↓
Benutzerfeedback auswerten



Inhalte und Abläufe kontinuierlich verbessern

Merksätze

“ Self-Service ersetzt nicht den Service Desk, sondern erweitert die Servicefähigkeit.

“ Wissen muss auffindbar, verständlich und aktuell sein.

“ Automatisierung braucht klare Verantwortung.

“ Ein automatisierter Fehler bleibt ein Fehler.

“ Jede gelöste Störung kann zukünftiges Wissen erzeugen.

“ Gute Standardisierung schafft Freiraum für komplexe Aufgaben.

Verwandte Seiten

- 3.1 Der Service Desk als zentraler Kontaktpunkt
 - 3.4 Tickets vollständig erfassen und kategorisieren
 - 3.6 Erstdiagnose, Lösung und funktionale Eskalation
 - 3.7 Ownership, hierarchische Eskalation und Major Incidents
 - 3.9 Kennzahlen, Qualität und Continual Improvement im Service Desk
 - Knowledge Management
 - Service Request Management
 - Change Enablement
 - Continual Improvement
-

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Desk
- PeopleCert – ITIL Practice Guide: Knowledge Management
- PeopleCert – ITIL Practice Guide: Service Request Management
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- ITIL Foundation – Version 5

Einordnung

Die dargestellten:

- Self-Service-Beispiele,
- Portalstrukturen,
- Knowledge-Artikel-Kriterien,
- Runbook-Beispiele,
- Automatisierungsregeln,
- Checklisten,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Portalstruktur,
- Knowledge-Artikel-Vorlage,
- Automatisierungsquote,
- Runbook-Form,
- KCS-Einführung,
- oder Self-Service-Pflicht

für alle Organisationen vor.

Die konkrete Ausgestaltung muss an:

- Services,
- Benutzergruppen,
- Sicherheitsanforderungen,
- technische Plattformen,
- Genehmigungswege,
- Service Levels,
- Fähigkeiten,
- und organisatorische Verantwortung

angepasst werden.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
