

4.3 Workarounds und Known Errors

“ Kurz erklärt

Ein **Workaround** ist eine vorübergehende Möglichkeit, die Auswirkungen eines Incidents oder Problems zu umgehen oder zu reduzieren.

Ein **Known Error** ist ein bekanntes Problem oder ein bekannter Fehlerzustand, bei dem Ursache, Zusammenhang oder Fehlerbild ausreichend verstanden sind.

Workarounds und Known Errors helfen dabei, Incidents schneller zu bearbeiten, Benutzer arbeitsfähig zu halten und Wissen für den Service Desk nutzbar zu machen.

Warum Workarounds wichtig sind

Nicht jede Ursache kann sofort dauerhaft beseitigt werden.

Gründe können sein:

- dauerhafte Lösung benötigt einen Change,
- Hersteller muss Fehler beheben,
- Wartungsfenster ist erforderlich,
- Risiko einer Sofortänderung ist zu hoch,
- Ursache ist noch nicht vollständig verstanden,
- Ersatzhardware ist nicht sofort verfügbar,
- Freigaben fehlen,
- oder eine technische Abhängigkeit muss zuerst geklärt werden.

Ein Workaround kann in solchen Situationen helfen, die Auswirkungen zu begrenzen.

Beispiele:

- alternativer Drucker,
- Webversion statt Desktop-Anwendung,
- Neustart eines betroffenen Dienstes,
- anderer Netzwerkzugang,
- manuelle Ersatzbearbeitung,
- temporäre Umleitung,
- Nutzung eines Ersatzsystems.

Ein Workaround ist nicht automatisch schlecht.

Er wird problematisch, wenn er dauerhaft genutzt wird und die eigentliche Ursache nie bearbeitet wird.

Workaround und dauerhafte Lösung unterscheiden

Workaround	Dauerhafte Lösung
reduziert oder umgeht Auswirkungen	beseitigt Ursache oder Risiko nachhaltig
oft kurzfristig verfügbar	benötigt häufig Analyse, Planung oder Change
kann Service schnell wieder nutzbar machen	verhindert Wiederholung besser
Ursache kann bestehen bleiben	Ursache wird behoben oder kontrolliert
muss dokumentiert und überwacht werden	muss getestet und dauerhaft übernommen werden

Beispiel:

Ein VPN-Client verliert nach dem Ruhezustand die Verbindung.

Workaround

VPN-Client vollständig beenden und neu starten.

Dauerhafte Lösung

Fehlerhafte Client-Version aktualisieren oder Konfiguration korrigieren.

Was ist ein Known Error?

Ein Known Error ist ein Problem oder Fehlerzustand, der bekannt und ausreichend beschrieben ist.

Typische Merkmale:

- Fehlerbild ist bekannt,
- betroffene Services oder Komponenten sind bekannt,
- Ursache oder wahrscheinliche Ursache ist dokumentiert,
- Auswirkungen sind beschrieben,
- Workaround ist vorhanden oder geprüft,
- dauerhafte Lösung ist geplant, offen oder bewusst zurückgestellt.

Ein Known Error muss nicht bedeuten, dass bereits eine endgültige Lösung vorhanden ist.

Gerade deshalb ist die Dokumentation wichtig.

Problem, Known Error, Workaround und Lösung unterscheiden

Begriff	Bedeutung
Problem	Ursache oder mögliche Ursache eines oder mehrerer Incidents
Known Error	bekanntes Problem oder bekannter Fehlerzustand mit dokumentiertem Wissen
Workaround	vorübergehende Maßnahme zur Umgehung oder Reduzierung der Auswirkungen
dauerhafte Lösung	Maßnahme zur Beseitigung der Ursache oder dauerhaften Risikoreduzierung

Beispiel:

Problem

Mehrere Benutzer verlieren regelmäßig die VPN-Verbindung.

Known Error

VPN-Client-Version 5.8 verursacht nach dem Ruhezustand fehlerhafte Tunnelzustände.

Workaround

VPN-Client vollständig beenden und neu starten.

Dauerhafte Lösung

Getestete Client-Version 5.9 über Change Enablement ausrollen.

Ein Workaround ist keine endgültige Lösung

Ein häufiger Fehler besteht darin, einen erfolgreichen Workaround als vollständige Lösung zu behandeln.

Beispiel:

Ein Dienst stürzt regelmäßig ab.

Der Service Desk startet den Dienst jedes Mal neu.

Dadurch ist der Service kurzfristig wieder verfügbar.

Die Ursache bleibt jedoch bestehen.

Mögliche offene Fragen:

- Warum stürzt der Dienst ab?

- Warum erkennt Monitoring den Fehler nicht früher?
- Warum gibt es keinen automatischen Neustart?
- Warum wurde die Ursache nicht untersucht?
- Warum tritt der Fehler regelmäßig auf?

“ **Merke**

Ein Workaround kann einen Incident lösen.

Er löst aber nicht automatisch das zugrunde liegende Problem.

Anforderungen an einen guten Workaround

Ein guter Workaround sollte:

- wirksam sein,
- verständlich beschrieben sein,
- reproduzierbar funktionieren,
- sicher sein,
- zulässig sein,
- keine größeren Folgerisiken erzeugen,
- für die Zielgruppe geeignet sein,
- dokumentiert sein,
- und regelmäßig überprüft werden.

Zu prüfen ist außerdem:

- Für welche Versionen gilt der Workaround?
- Für welche Benutzer oder Services gilt er?
- Welche Voraussetzungen bestehen?
- Welche Einschränkungen bleiben?
- Wie lange darf er genutzt werden?
- Wann muss eskaliert werden?
- Muss später eine Rücknahme erfolgen?

Ungeeignete Workarounds

Ein Workaround ist ungeeignet, wenn er:

- Sicherheitsvorgaben umgeht,
- Datenschutz verletzt,
- Datenverlust riskieren kann,
- unkontrollierte Änderungen verursacht,

- nur zufällig funktioniert,
- nicht dokumentiert ist,
- Benutzer überfordert,
- oder die Ursache verschlimmern kann.

Beispiel:

“ Benutzer sollen vertrauliche Dokumente über private E-Mail-Konten austauschen, weil der Dateiserver nicht erreichbar ist.

Das wäre technisch vielleicht möglich, aber organisatorisch und sicherheitlich ungeeignet.

Workaround dokumentieren

Ein Workaround sollte so dokumentiert sein, dass er wiederverwendbar ist.

Typische Inhalte:

- Titel,
- betroffener Service,
- Fehlerbild,
- Voraussetzungen,
- betroffene Versionen oder Komponenten,
- genaue Arbeitsschritte,
- erwartetes Ergebnis,
- Einschränkungen,
- Risiken,
- Abbruchkriterien,
- Eskalationshinweise,
- Gültigkeitsdauer,
- Verantwortlicher,
- Bezug zu Problem Record oder Known Error.

Eine ungenaue Notiz reicht nicht aus.

Ungeeignet:

“ Client neu starten.

Besser:

VPN-Client vollständig beenden, prüfen ob der Prozess beendet ist, Client neu starten und erneut verbinden. Gilt nur für Windows-Notebooks mit Client-Version 5.8 nach Ruhezustand. Wenn Verbindung weiterhin fehlschlägt, an Netzwerkteam eskalieren.

Known Error dokumentieren

Ein Known Error sollte mindestens enthalten:

- bekannte Symptome,
- betroffener Service,
- betroffene Configuration Items,
- betroffene Versionen,
- Ursache oder wahrscheinliche Ursache,
- Auswirkungen,
- Workaround,
- bekannte Einschränkungen,
- dauerhafte Lösung oder geplanter Change,
- Status,
- Verantwortlicher,
- Verknüpfung zu Incidents und Problems,
- Datum der letzten Prüfung.

Ziel ist, dass Service Desk und Fachgruppen schneller erkennen:

- ob ein neuer Incident zu einem bekannten Fehler passt,
- welcher Workaround genutzt werden kann,
- wann eskaliert werden muss,
- und ob eine dauerhafte Lösung geplant ist.

Known Error Database

Organisationen können bekannte Fehler in einer **Known Error Database** oder in einem vergleichbaren Wissenssystem dokumentieren.

Das Werkzeug ist nicht entscheidend.

Möglich sind beispielsweise:

- ITSM-System,
- Knowledge Base,
- Wiki,
- CMDB-Verknüpfung,
- Problem-Management-Modul,

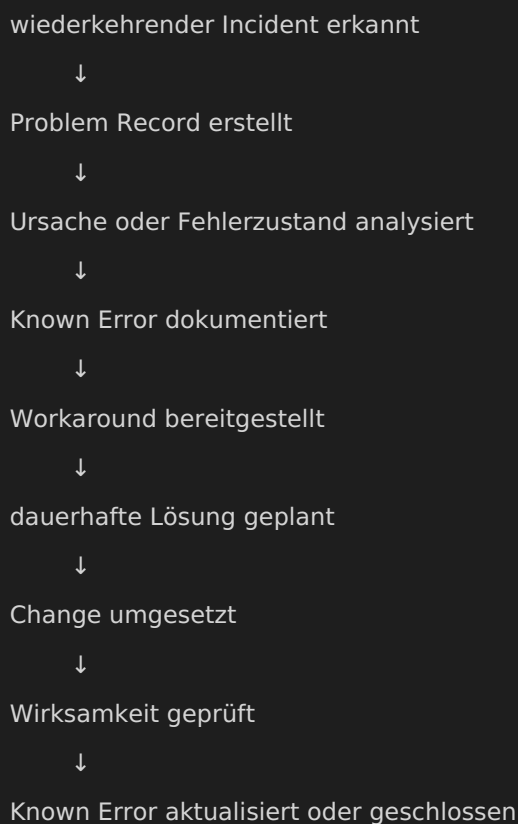
- Runbook-Sammlung.

Wichtig ist:

- Informationen sind auffindbar,
- Inhalte sind aktuell,
- Zielgruppen sind klar,
- Verknüpfungen zu Incidents und Problems sind vorhanden,
- und Workarounds sind verständlich beschrieben.

Lebenszyklus eines Known Errors

Ein möglicher Ablauf:



Nicht jeder Known Error folgt exakt diesem Ablauf.

Entscheidend ist, dass der Status nachvollziehbar bleibt.

Known Error offen lassen oder schließen

Ein Known Error sollte nicht einfach vergessen werden.

Mögliche Status:

- neu,
- in Analyse,
- Workaround verfügbar,
- dauerhafte Lösung geplant,
- Change in Vorbereitung,
- Lösung umgesetzt,
- Wirksamkeit wird geprüft,
- geschlossen,
- Risiko akzeptiert.

Ein Known Error kann offen bleiben, wenn:

- dauerhafte Lösung noch nicht verfügbar ist,
- Herstellerkorrektur aussteht,
- Änderung zu riskant wäre,
- Kosten unverhältnismäßig wären,
- oder Risiko bewusst akzeptiert wurde.

Dann müssen Workaround und Risiko weiterhin bekannt bleiben.

Workaround im Incident Management

Incident Management nutzt Workarounds, um den Service schnell wiederherzustellen.

Beispiele:

- Service Desk nutzt Known Error und Workaround,
- Benutzer erhält eine sichere Zwischenlösung,
- Incident wird mit Bezug zum Known Error dokumentiert,
- Problem Management verfolgt die Ursache weiter.

Wichtig:

Der Incident kann abgeschlossen werden, wenn der Benutzer wieder arbeitsfähig ist und die Organisation dies so definiert.

Das Problem kann trotzdem offen bleiben.

Workaround im Problem Management

Problem Management verwendet Workarounds, um Auswirkungen zu reduzieren, während die Ursache weiter untersucht wird.

Aufgaben:

- Workaround prüfen,

- Risiken bewerten,
- Dokumentation erstellen,
- Service Desk informieren,
- Known Error anlegen,
- dauerhafte Lösung vorbereiten,
- Change-Bedarf klären,
- Wirksamkeit überwachen.

Ein Workaround ist oft ein wichtiger Zwischenschritt zwischen Analyse und dauerhafter Lösung.

Workaround und Change Enablement

Ein Workaround kann selbst eine Änderung enthalten.

Beispiele:

- DNS-Eintrag temporär ändern,
- Dienst auf Ersatzsystem umleiten,
- Firewall-Regel vorübergehend anpassen,
- Konfiguration zurücksetzen,
- Zertifikat austauschen,
- Softwareversion zurückrollen.

Solche Maßnahmen müssen nach Risiko bewertet werden.

Nicht jeder Workaround darf ohne Change-Steuerung durchgeführt werden.

Die Organisation muss festlegen, welche Workarounds:

- als Standardmaßnahme erlaubt sind,
- eine Genehmigung benötigen,
- nur im Notfall zulässig sind,
- oder über einen Emergency Change laufen müssen.

Workaround und Knowledge Management

Workarounds sind nur dann wertvoll, wenn sie gefunden und verstanden werden.

Deshalb sollten sie in Knowledge Management eingebunden werden.

Zu klären ist:

- Wer darf den Artikel sehen?
- Ist der Artikel für Benutzer oder nur für IT?
- Welche Begriffe nutzen Benutzer bei der Suche?
- Muss es eine Kurzversion für den Service Desk geben?

- Gibt es Sicherheits- oder Datenschutzhinweise?
- Wann wird der Artikel überprüft?
- Wer ist für Aktualisierung verantwortlich?

Ein Workaround, der nur in einem alten Ticket steht, hilft beim nächsten Incident kaum.

Workaround und Service Desk

Der Service Desk benötigt klare Informationen:

- Wann passt dieser Workaround?
- Wann passt er nicht?
- Welche Fragen müssen gestellt werden?
- Welche Schritte sind erlaubt?
- Welche Risiken bestehen?
- Wann muss eskaliert werden?
- Wie wird der Incident dokumentiert?
- Welche Benutzerinformation ist freigegeben?

Beispiel:

“ Wenn Benutzer nach Ruhezustand keine VPN-Verbindung herstellen können und Client-Version 5.8 verwendet wird, Workaround aus Artikel KB-023 anwenden. Bei anderer Version oder Fehlermeldung an Netzwerkteam eskalieren.

Workaround und Benutzerkommunikation

Benutzer sollten verständlich informiert werden.

Wichtig ist:

- klar sagen, dass es eine Zwischenlösung ist,
- keine unsicheren technischen Vermutungen als Ursache darstellen,
- Einschränkungen nennen,
- nächsten Schritt erklären,
- bei Bedarf Statusmeldung ankündigen.

Beispiel:

“ Für diesen Fehler gibt es aktuell eine sichere Zwischenlösung. Bitte starten Sie den VPN-Client vollständig neu und verbinden Sie sich anschließend erneut. Die

Ursache wird weiter untersucht. Eine dauerhafte Korrektur ist in Vorbereitung.

Ungeeignet:

“ Das ist ein bekannter Bug, machen Sie einfach Neustart.

Risiken dauerhaft genutzter Workarounds

Wenn Workarounds dauerhaft genutzt werden, entstehen Risiken:

- Ursache bleibt bestehen,
- Benutzer gewöhnen sich an Umwege,
- manuelle Fehler nehmen zu,
- Supportaufwand bleibt hoch,
- Sicherheitsrisiken werden akzeptiert,
- technische Schulden wachsen,
- und echte Verbesserungen werden verschoben.

Ein Workaround sollte deshalb regelmäßig überprüft werden.

Fragen:

- Wird der Workaround noch gebraucht?
- Gibt es inzwischen eine dauerhafte Lösung?
- Wie oft wird er genutzt?
- Welche Risiken bestehen?
- Gibt es Beschwerden?
- Ist ein Change geplant?
- Wurde Risiko bewusst akzeptiert?

Known Error und Risikoakzeptanz

Nicht jeder Known Error wird sofort beseitigt.

Beispiel:

Ein selten auftretender Fehler betrifft eine alte Anwendung.

Der Workaround funktioniert zuverlässig.

Eine dauerhafte Lösung wäre sehr teuer und riskant.

Die Organisation kann entscheiden, das Risiko vorerst zu akzeptieren.

Dann sollte dokumentiert werden:

- wer das Risiko akzeptiert,
 - warum es akzeptiert wird,
 - welche Einschränkungen bestehen,
 - wie lange die Akzeptanz gilt,
 - wann neu bewertet wird,
 - und welcher Workaround genutzt wird.
-

Known Error bei Lieferantenfehlern

Ein Known Error kann auch durch einen Hersteller oder Lieferanten bestätigt werden.

Beispiele:

- Software-Bug,
- Firmwarefehler,
- Cloud-Service-Problem,
- Treiberproblem,
- API-Verhalten,
- Inkompatibilität.

Zu dokumentieren ist:

- Herstellerreferenz,
- betroffene Version,
- bestätigte Symptome,
- empfohlener Workaround,
- erwartete Korrektur,
- Lieferantenticket,
- interne Auswirkungen,
- geplanter interner Umgang.

Auch wenn ein Lieferant verantwortlich ist, bleibt die interne Serviceverantwortung bestehen.

Known Error und Major Incident

Nach einem Major Incident kann ein Known Error entstehen.

Beispiel:

Ein zentraler Dienst fällt wegen einer bestimmten Datenbankversion aus.

Der Service wurde wiederhergestellt.

Die dauerhafte Korrektur benötigt aber ein geplantes Upgrade.

Bis dahin wird ein Known Error dokumentiert mit:

- Symptomen,
- betroffenen Services,
- Workaround,
- Monitoringhinweisen,
- Eskalationsregeln,
- geplanter dauerhafter Lösung.

Dadurch kann bei Wiederholung schneller reagiert werden.

Beispiel: VPN-Client

Symptom

Benutzer verlieren nach dem Ruhezustand die VPN-Verbindung.

Problem

Mehrere Incidents mit gleichem Muster.

Known Error

VPN-Client-Version 5.8 verursacht nach Ruhezustand fehlerhafte Tunnelzustände.

Workaround

Client vollständig beenden und neu starten.

Dauerhafte Lösung

Version 5.9 testen und ausrollen.

Knowledge

Service Desk erhält einen Artikel mit Prüfschritten, Workaround und Eskalationskriterien.

Beispiel: Druckerwarteschlange

Symptom

Etiketten werden nicht gedruckt.

Problem

Druckwarteschlange blockiert regelmäßig durch bestimmte PDF-Dateien.

Known Error

Der aktuell installierte Druckertreiber verarbeitet bestimmte PDF-Formate fehlerhaft.

Workaround

Fehlerhaften Auftrag entfernen und Datei über alternative Druckoption ausgeben.

Dauerhafte Lösung

Treiberupdate testen und über Change Enablement ausrollen.

Beispiel: Zertifikat

Symptom

Anmeldung an Fachanwendung schlägt fehl.

Problem

Zertifikat des Anmeldedienstes ist abgelaufen.

Known Error

Zertifikatsablauf wird nicht überwacht und kann zu Anmeldeausfällen führen.

Workaround

Kein vollwertiger Workaround verfügbar.

Dauerhafte Lösung

Zertifikat erneuern, Monitoring einführen, Owner festlegen und Runbook erstellen.

“ Merke

Nicht jeder Known Error besitzt einen geeigneten Workaround.

Beispiel: Speicher läuft voll

Symptom

Server kann keine neuen Daten schreiben.

Problem

Logdateien wachsen unkontrolliert.

Known Error

Logrotation ist fehlerhaft konfiguriert.

Workaround

Temporäre Bereinigung nach Runbook durchführen.

Dauerhafte Lösung

Logrotation korrigieren, Monitoring-Grenzen anpassen und Verantwortlichkeit festlegen.

Typische Fehler

Fehler 1

Workaround wird als dauerhafte Lösung behandelt.

Fehler 2

Known Error wird nicht dokumentiert.

Fehler 3

Workaround steht nur in einem einzelnen Ticket.

Fehler 4

Service Desk kennt den Workaround nicht.

Fehler 5

Benutzer erhalten unsichere oder unklare Anweisungen.

Fehler 6

Workaround umgeht Sicherheitsregeln.

Fehler 7

Risiken und Einschränkungen werden nicht genannt.

Fehler 8

Known Errors bleiben dauerhaft offen, ohne Neubewertung.

Fehler 9

Dauerhafte Lösung wird nicht geplant oder verfolgt.

Fehler 10

Lieferantenfehler werden nicht intern dokumentiert.

Fehler 11

Workaround wird nicht getestet.

Fehler 12

Nach Umsetzung der dauerhaften Lösung wird der Knowledge-Artikel nicht aktualisiert.

Checkliste Workaround erstellen

- ☐ Fehlerbild eindeutig beschrieben
- ☐ betroffener Service genannt
- ☐ Voraussetzungen dokumentiert
- ☐ genaue Schritte beschrieben
- ☐ erwartetes Ergebnis beschrieben
- ☐ Risiken bewertet
- ☐ Einschränkungen genannt
- ☐ Abbruchkriterien definiert
- ☐ Eskalationsweg beschrieben
- ☐ Zielgruppe festgelegt
- ☐ Verantwortlicher benannt
- ☐ regelmäßige Prüfung vorgesehen

Checkliste Known Error dokumentieren

- ☐ Problem oder Fehlerzustand beschrieben
- ☐ betroffene Services und Komponenten genannt
- ☐ bekannte Symptome dokumentiert
- ☐ Ursache oder wahrscheinliche Ursache beschrieben
- ☐ Auswirkungen bewertet
- ☐ zugehörige Incidents verknüpft
- ☐ Workaround vorhanden oder als nicht verfügbar gekennzeichnet
- ☐ dauerhafte Lösung oder nächster Schritt dokumentiert
- ☐ Status festgelegt
- ☐ Verantwortlicher benannt
- ☐ Knowledge-Artikel oder Runbook verknüpft
- ☐ Datum der nächsten Überprüfung festgelegt

Checkliste Workaround im Betrieb nutzen

- ☐ passt das aktuelle Symptom zum Known Error?
- ☐ sind Version, Service oder Komponente betroffen?
- ☐ ist der Workaround für diesen Fall zulässig?
- ☐ wurden Risiken geprüft?
- ☐ wurde der Benutzer verständlich informiert?
- ☐ wurde die Anwendung dokumentiert?
- ☐ muss funktional eskaliert werden?
- ☐ muss Problem Management informiert werden?
- ☐ ist eine dauerhafte Lösung geplant?
- ☐ wurde die Wirksamkeit bestätigt?

Checkliste Known Error schließen

- ☐ dauerhafte Lösung umgesetzt
- ☐ Wirksamkeit geprüft
- ☐ zugehörige Incidents berücksichtigt
- ☐ Knowledge-Artikel aktualisiert
- ☐ Workaround entfernt oder angepasst

- ☐ Monitoring oder Runbook aktualisiert
 - ☐ Rest Risiko bewertet
 - ☐ Stakeholder informiert
 - ☐ Problem Record aktualisiert
 - ☐ Lessons Learned dokumentiert
-

Bedeutung für Fachinformatiker für Systemintegration

Für Fachinformatiker ist der saubere Umgang mit Workarounds und Known Errors besonders wichtig.

Im Arbeitsalltag bedeutet das:

- wiederkehrende Fehler erkennen,
- wirksame Zwischenlösungen sauber dokumentieren,
- Risiken eines Workarounds bewerten,
- bekannte Fehler für den Service Desk nutzbar machen,
- dauerhafte Lösungen vorbereiten,
- Changes fachlich unterstützen,
- und Knowledge-Artikel aktuell halten.

Ein guter Workaround spart Zeit.

Ein gut dokumentierter Known Error verhindert doppelte Analyse.

Eine dauerhafte Lösung verbessert den Service nachhaltig.

Zusammenfassung

“ Incident oder wiederkehrendes Fehlerbild erkennen

↓

Problem oder mögliche Ursache untersuchen

↓

Known Error dokumentieren

↓

sicheren Workaround bereitstellen

↓

Service Desk und Benutzer angemessen informieren

↓

dauerhafte Lösung planen

↓

Change bei Bedarf durchführen



Wirksamkeit prüfen



Known Error und Knowledge aktualisieren oder schließen

Merksätze

“ Ein Workaround ist hilfreich, aber nicht automatisch eine Lösung.

“ Ein Known Error muss auffindbar und verständlich dokumentiert sein.

“ Nicht jeder Known Error besitzt einen sicheren Workaround.

“ Workarounds dürfen Sicherheitsregeln nicht aushebeln.

“ Dauerhaft genutzte Workarounds sind ein Hinweis auf ungelöste Probleme.

“ Gute Known-Error-Dokumentation macht zukünftige Incidents schneller lösbar.

Verwandte Seiten

- 4.1 Problem Management – Ziele, Begriffe und Abgrenzung
- 4.2 Ursachenanalyse
- 4.4 Trendanalyse und proaktives Problem Management
- 4.5 Problem Management im Zusammenspiel mit Incident, Change und Knowledge Management
- 3.6 Erstdiagnose, Lösung und funktionale Eskalation
- Knowledge Management
- Change Enablement

- Continual Improvement
-

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Problem Management
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Knowledge Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- ITIL Foundation – Version 5

Einordnung

Die dargestellten:

- Workaround-Beispiele,
- Known-Error-Strukturen,
- Checklisten,
- Statusmodelle,
- Dokumentationshinweise,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Known-Error-Datenbank-Struktur,
- Workaround-Vorlage,
- Statusliste,
- Dokumentationstiefe,
- Freigaberegeln,
- oder Toolauswahl

für alle Organisationen vor.

Die konkrete Umsetzung muss an:

- Services,
- Risiken,
- Sicherheitsanforderungen,
- Supportmodell,
- Knowledge Management,
- Change Enablement,
- Lieferanten,
- und organisatorische Verantwortlichkeiten

angepasst werden.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
