

7.3 Known Errors, Workarounds und Wissensnutzung im Service Desk

“ Kurz erklärt

Known Errors und Workarounds sind besonders wichtig für den Service Desk.

Sie helfen dabei, bekannte Fehlerbilder schneller zu erkennen, Benutzer schneller wieder arbeitsfähig zu machen und wiederkehrende Incidents einheitlich zu bearbeiten.

Entscheidend ist, dass dieses Wissen verständlich dokumentiert, leicht auffindbar, aktuell und mit Incident-, Problem- und Knowledge Management verknüpft ist.

Warum Known Errors und Workarounds im Service Desk wichtig sind

Der Service Desk ist häufig die erste Stelle, die wiederkehrende Fehler bemerkt.

Beispiele:

- mehrere Benutzer melden denselben VPN-Fehler,
- eine Anwendung zeigt nach einem Update immer dieselbe Fehlermeldung,
- ein Druckertreiber verursacht wiederkehrende Probleme,
- ein Zertifikat läuft ab,
- ein Workaround wird regelmäßig angewendet,
- ein bekannter Fehler tritt nach bestimmten Bedingungen auf.

Ohne dokumentiertes Wissen muss der Service Desk jeden Fall neu analysieren.

Mit gut gepflegten Known Errors und Workarounds kann er schneller reagieren.

Known Error

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

Ein Known Error kann enthalten:

- bekanntes Symptom,
- betroffener Service,

- betroffene Version,
- Ursache oder wahrscheinliche Ursache,
- Workaround,
- dauerhafte Lösung,
- Status,
- Risiko,
- Eskalationsweg,
- Verknüpfung zu Problem Record,
- Verknüpfung zu Knowledge-Artikel.

Ein Known Error muss nicht bedeuten, dass die endgültige Lösung bereits umgesetzt wurde.

Workaround

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

Beispiele:

- VPN-Client vollständig neu starten,
- alternative Webversion nutzen,
- Druckauftrag über anderen Drucker ausgeben,
- Dienst kontrolliert neu starten,
- temporäre Bereinigung nach Runbook durchführen,
- anderes Anmeldeverfahren verwenden,
- bekannte fehlerhafte Funktion vorübergehend vermeiden.

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

Known Error und Workaround unterscheiden

Begriff	Bedeutung	Beispiel
Known Error	bekannter Fehlerzustand oder bekanntes Problem	VPN-Client-Version 5.8 verliert nach Ruhezustand Verbindung
Workaround	vorübergehende Maßnahme zur Umgehung	VPN-Client vollständig beenden und neu starten
dauerhafte Lösung	beseitigt Ursache oder reduziert Risiko dauerhaft	neue VPN-Client-Version ausrollen
Knowledge-Artikel	nutzbare Dokumentation für Zielgruppe	Service-Desk-Artikel mit Symptomen und Prüfschritten

Warum Workarounds dokumentiert werden müssen

Ein Workaround, der nur einzelnen Personen bekannt ist, hilft nicht zuverlässig.

Typische Folgen fehlender Dokumentation:

- Service Desk arbeitet uneinheitlich,
- Benutzer erhalten unterschiedliche Antworten,
- Incidents dauern länger,
- gleiche Analyse wird mehrfach durchgeführt,
- Risiken werden übersehen,
- Eskalationen erfolgen zu spät,
- Problem Management erkennt Muster schlechter,
- neue Mitarbeitende müssen Wissen mündlich erfragen.

Ein guter Workaround muss für die richtige Zielgruppe auffindbar und verständlich sein.

Ein Workaround ist keine endgültige Lösung

Ein häufiger Fehler ist, einen Workaround als Lösung zu behandeln.

Beispiel:

Ein Dienst stürzt regelmäßig ab.

Der Service Desk startet den Dienst jedes Mal neu.

Dadurch funktioniert der Service kurzfristig wieder.

Die eigentliche Ursache bleibt aber bestehen.

Offene Fragen:

- Warum stürzt der Dienst ab?
- Warum erkennt Monitoring den Fehler nicht früher?
- Warum ist kein automatischer Neustart vorgesehen?
- Warum tritt der Fehler regelmäßig auf?
- Gibt es ein Problem Record?
- Gibt es eine geplante dauerhafte Lösung?

“ Merke

Ein Workaround kann einen Incident abschließen.

Das zugrunde liegende Problem kann trotzdem offen bleiben.

Wissensnutzung im Service Desk

Der Service Desk sollte Knowledge aktiv nutzen.

Beispiele:

- bei Ticketerstellung nach passenden Artikeln suchen,
- bekannte Symptome mit Known Errors vergleichen,
- Workaround nach geprüfter Anleitung anwenden,
- Ticket mit Knowledge-Artikel verknüpfen,
- fehlende oder falsche Artikel melden,
- wiederkehrende Incidents an Problem Management geben,
- Benutzer mit freigegebenen Informationen versorgen.

Knowledge Management ist nur wirksam, wenn das Wissen im Alltag verwendet wird.

Service-Desk-Artikel zu Known Errors

Ein guter Service-Desk-Artikel zu einem Known Error enthält:

- eindeutigen Titel,
- betroffenen Service,
- bekannte Symptome,
- betroffene Versionen oder Komponenten,
- Prüfschritte,
- Workaround,
- Risiken und Einschränkungen,
- wann der Artikel nicht gilt,
- Eskalationskriterien,
- Verknüpfung zum Problem Record,
- geplante dauerhafte Lösung,
- Review-Datum,
- Owner.

Der Artikel muss so geschrieben sein, dass der Service Desk ihn unter Zeitdruck nutzen kann.

Beispielstruktur für einen Known-Error-Artikel

Abschnitt	Inhalt
Titel	VPN verbindet nach Ruhezustand nicht mehr
Betroffener Service	VPN-Zugang
Symptom	Verbindung bricht nach Standby ab
Betroffen	Windows-Notebooks mit Client-Version 5.8

Abschnitt	Inhalt
Prüfung	Clientversion und Fehlerbild abgleichen
Workaround	VPN-Client vollständig beenden und neu starten
Eskalation	bei anderer Version an Netzwerkteam
Problem Record	verknüpft
Dauerhafte Lösung	Rollout Version 5.9 geplant
Review	nach Rollout prüfen

Titel und Suchbegriffe

Known-Error-Artikel müssen über typische Suchbegriffe gefunden werden.

Ungeeignet:

“ Fehlercode 0x802-VPN-Tunnelstate

Besser:

“ VPN verbindet nach Ruhezustand nicht mehr

Ungeeignet:

“ Problem mit Auth-Komponente

Besser:

“ Anmeldung im Mitarbeiterportal schlägt fehl

Ein guter Titel nutzt die Sprache, die im Ticket oder im Benutzerkontakt tatsächlich vorkommt.

Symptome klar beschreiben

Der Service Desk muss schnell erkennen können, ob ein neuer Incident zum Known Error passt.

Gute Symptombeschreibungen enthalten:

- sichtbare Fehlermeldung,
- Zeitpunkt oder Auslöser,
- betroffene Anwendung,
- betroffene Version,
- betroffene Benutzergruppe,
- betroffene Umgebung,
- typische Begleitumstände,
- bekannte Ausschlusskriterien.

Beispiel:

“ Fehler tritt nur nach Ruhezustand auf Windows-Notebooks mit VPN-Client-Version 5.8 auf. Wenn der Fehler direkt nach Neustart oder bei anderer Version auftritt, gilt dieser Known Error nicht automatisch.

Prüfschritte

Prüfschritte helfen, den Known Error sicher zu erkennen.

Beispiele:

- Service prüfen,
- Version prüfen,
- Benutzergruppe prüfen,
- Fehlermeldung vergleichen,
- Zeitpunkt prüfen,
- letzte Changes prüfen,
- Known-Error-Status prüfen,
- Monitoring prüfen,
- Ticketverlauf prüfen.

Prüfschritte sollten kurz, eindeutig und reproduzierbar sein.

Ungeeignet:

“ Prüfen, ob es das bekannte Problem ist.

Besser:

VPN-Client-Version prüfen. Gilt nur für Version 5.8. Fehler tritt nach Ruhezustand auf. Wenn Version abweicht, an Netzwerkteam eskalieren.

Workaround-Schritte

Ein Workaround muss so beschrieben sein, dass er sicher ausgeführt werden kann.

Er sollte enthalten:

- genaue Schritte,
- Voraussetzungen,
- erwartetes Ergebnis,
- Risiken,
- Einschränkungen,
- Abbruchkriterien,
- Eskalation bei Fehlschlag,
- Hinweis zur Dokumentation im Ticket.

Ungeeignet:

“ Client neu starten.

Besser:

“ VPN-Client vollständig beenden, prüfen ob kein VPN-Prozess mehr läuft, Client erneut starten und Verbindung testen. Wenn Verbindung weiterhin fehlschlägt, Ticket an Netzwerkteam eskalieren.

Wann der Workaround nicht angewendet werden darf

Ein guter Artikel beschreibt auch Grenzen.

Beispiele:

- gilt nur für bestimmte Version,
- gilt nicht bei Sicherheitsvorfall,
- gilt nicht bei Datenverlust,
- gilt nicht bei produktivem Datenbankfehler,
- gilt nicht außerhalb eines genehmigten Wartungsfensters,

- gilt nicht ohne Freigabe,
- gilt nicht bei kritischen Services ohne Eskalation.

Diese Grenzen verhindern, dass ein Workaround falsch oder riskant angewendet wird.

Risiken eines Workarounds

Workarounds können Risiken haben.

Beispiele:

- Datenverlust,
- Sicherheitsrisiko,
- Unterbrechung für andere Benutzer,
- falsche Berechtigungen,
- Folgefehler,
- manuelle Fehler,
- unvollständige Wiederherstellung,
- Verstoß gegen Vorgaben,
- verdeckte Ursache bleibt bestehen.

Deshalb sollten Risiken im Artikel genannt werden.

Ein Workaround ohne Risikohinweis kann zu falscher Anwendung führen.

Eskalationskriterien

Ein Service-Desk-Artikel sollte klar sagen, wann eskaliert werden muss.

Beispiele:

- Workaround schlägt fehl,
- Fehlerbild passt nicht vollständig,
- anderer Service ist betroffen,
- mehrere Benutzer oder Standort betroffen,
- Sicherheitsbezug möglich,
- kritischer Service betroffen,
- Datenverlust möglich,
- gleiche Störung wiederholt sich häufig,
- Benutzer hat Sonderrolle,
- bekannte Version stimmt nicht.

Eskalationskriterien verhindern unnötige Verzögerungen.

Ticketdokumentation

Wenn ein Known Error oder Workaround genutzt wurde, sollte das Ticket sauber dokumentiert werden.

Wichtige Angaben:

- welcher Knowledge-Artikel genutzt wurde,
- welcher Known Error vermutet oder bestätigt wurde,
- welche Prüfschritte durchgeführt wurden,
- ob der Workaround funktioniert hat,
- welche Version betroffen war,
- ob eskaliert wurde,
- ob Benutzer informiert wurde,
- ob Folgeproblem erkennbar ist.

Gute Ticketdokumentation unterstützt Trendanalyse und Problem Management.

Verknüpfung mit Problem Management

Known Errors entstehen oft aus Problem Management.

Problem Management liefert:

- Ursache oder wahrscheinliche Ursache,
- beitragende Faktoren,
- bekannte Symptome,
- Workaround,
- dauerhafte Lösung,
- Risiko,
- Status,
- geplante Changes.

Der Service Desk nutzt dieses Wissen im Incident Management.

Neue Incidents liefern wiederum Informationen zurück an Problem Management.

Wann der Service Desk Problem Management informieren sollte

Problem Management sollte informiert werden, wenn:

- ein Workaround sehr häufig genutzt wird,
- ein Known Error häufiger auftritt als erwartet,
- neue Symptome auftreten,
- ein bekannter Workaround nicht mehr funktioniert,
- ein neuer wiederkehrender Fehler sichtbar wird,
- ein Major Incident droht,

- viele Benutzer betroffen sind,
- Service Desk auffällige Trends erkennt.

Der Service Desk ist eine wichtige Frühwarnquelle.

Verknüpfung mit Change Enablement

Eine dauerhafte Lösung für einen Known Error benötigt häufig einen Change.

Beispiele:

- Softwareupdate,
- Konfigurationsänderung,
- Treiberupdate,
- Zertifikatserneuerung,
- Firewall-Anpassung,
- Monitoring-Erweiterung,
- Berechtigungsänderung,
- Austausch einer Komponente.

Der Knowledge-Artikel sollte zeigen:

- ob ein Change geplant ist,
 - wann eine dauerhafte Lösung erwartet wird,
 - ob der Workaround bis dahin gilt,
 - wann der Artikel überprüft werden muss.
-

Verknüpfung mit Release Management

Nach einem Release kann sich der Status eines Known Errors ändern.

Beispiele:

- Known Error wurde behoben,
- Workaround ist nicht mehr gültig,
- neue Version hat anderes Fehlerbild,
- Artikel benötigt neue Screenshots,
- Service Desk braucht neue Prüfschritte,
- Benutzerinformation muss angepasst werden.

Nach Releases sollten relevante Knowledge-Artikel geprüft werden.

Verknüpfung mit Service Configuration Management

Known Errors und Workarounds sollten mit Services und CIs verknüpft werden.

Beispiele:

- Service: VPN-Zugang,
- CI: VPN-Client-Version 5.8,
- CI: VPN-Gateway,
- CI: Identity Provider,
- CI: Zertifikat,
- Problem Record,
- Change Record.

Vorteile:

- Artikel werden im Ticket besser vorgeschlagen,
- betroffene Versionen sind erkennbar,
- Trendanalyse wird besser,
- Changes können Artikel gezielt aktualisieren,
- Service Desk findet schneller passende Informationen.

Benutzerkommunikation bei Known Errors

Benutzer sollten verständlich informiert werden.

Gute Kommunikation enthält:

- was betroffen ist,
- welche Zwischenlösung verfügbar ist,
- was Benutzer tun sollen,
- welche Einschränkung besteht,
- ob an einer dauerhaften Lösung gearbeitet wird,
- wo weitere Informationen stehen.

Ungeeignet:

“ Das ist ein bekannter Bug, einfach neu starten.

Besser:

“ Für dieses VPN-Problem gibt es aktuell eine sichere Zwischenlösung. Bitte schließen Sie den VPN-Client vollständig und starten Sie ihn erneut. Die Ursache ist bekannt, eine dauerhafte Korrektur wird vorbereitet.

Interne und externe Artikel trennen

Ein Known Error kann verschiedene Artikel benötigen.

Benutzerartikel

- einfache Erklärung,
- sichere Handlung,
- Supportweg,
- keine internen Details.

Service-Desk-Artikel

- Symptome,
- Prüfschritte,
- Workaround,
- Eskalation,
- Ticketdokumentation.

Fachteam-Artikel

- technische Ursache,
- Logs,
- Konfiguration,
- dauerhafte Lösung,
- Change-Planung.

Nicht jedes Detail gehört in jeden Artikel.

Known Error ohne Workaround

Nicht jeder Known Error besitzt einen sicheren Workaround.

Beispiele:

- Zertifikat ist abgelaufen,
- Datenbank ist beschädigt,
- Sicherheitslücke wird aktiv ausgenutzt,
- zentraler Dienst ist ausgefallen,
- Datenverlust droht.

Dann muss der Artikel klar sagen:

- kein Workaround verfügbar,
- sofort eskalieren,
- Benutzer nicht zu unsicheren Ersatzwegen anleiten,

- Statuskommunikation nutzen,
- Problem oder Major Incident prüfen.

Ein fehlender Workaround sollte ausdrücklich dokumentiert werden.

Known Error mit Risikoakzeptanz

Manche Known Errors bleiben längere Zeit offen.

Gründe:

- dauerhafte Lösung ist teuer,
- Herstellerkorrektur steht aus,
- Änderung ist riskant,
- Workaround funktioniert ausreichend,
- Service wird bald abgelöst,
- Risiko wurde bewusst akzeptiert.

Dann sollte dokumentiert werden:

- wer das Risiko akzeptiert hat,
 - wie lange die Akzeptanz gilt,
 - welcher Workaround erlaubt ist,
 - wann neu bewertet wird,
 - welche Einschränkungen bestehen.
-

Knowledge-Artikel im Ticket vorschlagen

Viele ITSM-Systeme können Artikel im Ticket vorschlagen.

Grundlagen dafür:

- guter Titel,
- passende Suchbegriffe,
- Servicebezug,
- Kategorie,
- CI-Verknüpfung,
- Known-Error-Verknüpfung,
- klare Symptome,
- aktuelle Inhalte.

Automatische Vorschläge helfen nur, wenn die Artikelqualität gut ist.

Schlechte Vorschläge führen dazu, dass Mitarbeitende die Funktion ignorieren.

Nutzung messen

Die Nutzung von Known-Error- und Workaround-Artikeln sollte ausgewertet werden.

Mögliche Fragen:

- Welche Artikel werden häufig genutzt?
- Welche Workarounds werden besonders oft angewendet?
- Welche Artikel lösen Incidents erfolgreich?
- Wo brechen Mitarbeitende ab?
- Welche Artikel werden schlecht bewertet?
- Welche Suchanfragen führen zu keinem Treffer?
- Welche Known Errors erzeugen weiterhin viele Tickets?

Diese Daten helfen, Wissen und Problems gezielt zu verbessern.

Artikelqualität aus Service-Desk-Sicht

Ein Service-Desk-Artikel ist gut, wenn er:

- schnell auffindbar ist,
- klare Symptome nennt,
- Prüfschritte enthält,
- Workaround sicher beschreibt,
- Eskalationsregeln nennt,
- keine unnötige Fachsprache nutzt,
- Risiken sichtbar macht,
- mit Service oder CI verknüpft ist,
- aktuell ist,
- und im Ticket nutzbar dokumentiert werden kann.

Der Artikel muss im Arbeitsfluss helfen, nicht nur formal existieren.

Feedback aus dem Service Desk

Der Service Desk sollte Feedback zu Artikeln geben können.

Beispiele:

- Artikel nicht gefunden,
- Titel unklar,
- Schritt fehlt,
- Workaround funktioniert nicht mehr,
- Fehlerbild hat sich geändert,
- Screenshot veraltet,

- Eskalationsweg falsch,
- Benutzer verstehen Anleitung nicht,
- Artikel ist doppelt vorhanden.

Feedback muss bearbeitet werden.

Sonst verliert der Service Desk Vertrauen in die Knowledge Base.

Wissenspflege nach gelösten Incidents

Nach gelösten Incidents sollte geprüft werden:

- Gibt es bereits einen passenden Artikel?
- Muss der Artikel ergänzt werden?
- War der Artikel korrekt?
- Sollte ein neuer Artikel entstehen?
- Ist ein Known Error erkennbar?
- Gibt es einen neuen Workaround?
- Sollte Problem Management informiert werden?

Diese Prüfung muss nicht bei jedem kleinen Ticket ausführlich sein.

Bei wiederkehrenden oder auffälligen Fällen ist sie sehr wertvoll.

Wissenspflege nach Problem-Abschluss

Nach Problem Management sollten relevante Artikel aktualisiert werden.

Zu prüfen ist:

- wurde ein Known Error erstellt,
- wurde ein Workaround dokumentiert,
- wurde eine dauerhafte Lösung umgesetzt,
- ist der Workaround noch gültig,
- muss der Service Desk neue Prüfschritte kennen,
- muss ein Benutzerartikel erstellt werden,
- müssen alte Artikel archiviert werden.

Problem-Abschluss ohne Knowledge-Prüfung führt dazu, dass Wissen im Betrieb nicht ankommt.

Wissenspflege nach Change oder Release

Nach Changes und Releases muss geprüft werden:

- ist der Known Error behoben,

- ist der Workaround noch gültig,
- haben sich Versionen geändert,
- haben sich Menüpfade geändert,
- haben sich Symptome geändert,
- muss der Artikel archiviert werden,
- muss der Service Desk informiert werden,
- müssen Benutzerinformationen angepasst werden.

Ein veralteter Workaround kann nach einem Release neue Probleme verursachen.

Praxisbeispiel: VPN Known Error

Situation

Nach dem Ruhezustand verlieren Windows-Notebooks die VPN-Verbindung.

Known Error

VPN-Client-Version 5.8 verliert nach Ruhezustand den Tunnelzustand.

Service-Desk-Artikel

- Fehlerbild,
- betroffene Version,
- Prüfschritte,
- Workaround,
- Eskalation bei anderer Version,
- Problem Record,
- geplanter Change auf Version 5.9.

Nutzen

Der Service Desk erkennt den Fall schneller und dokumentiert ihn einheitlich.

Praxisbeispiel: Druckertreiber

Situation

Etikettendruck funktioniert bei bestimmten PDF-Dateien nicht.

Known Error

Der installierte Treiber verarbeitet bestimmte PDF-Formate fehlerhaft.

Workaround

Fehlerhaften Druckauftrag entfernen und alternative Druckoption nutzen.

Dauerhafte Lösung

Treiberupdate testen und ausrollen.

Knowledge-Nutzen

Service Desk kann betroffene Fälle schneller erkennen und vermeidet unnötige Hardwaretauschversuche.

Praxisbeispiel: Kein Workaround verfügbar

Situation

Anmeldung an einer Fachanwendung schlägt wegen abgelaufenem Zertifikat fehl.

Known Error

Zertifikat des Anmeldedienstes ist abgelaufen.

Workaround

Kein sicherer Workaround verfügbar.

Service-Desk-Anweisung

- Incident priorisieren,
- an Plattform-Team eskalieren,
- Statusmeldung verwenden,
- Benutzer nicht auf unsichere Ersatzwege verweisen.

Nutzen

Der Service Desk reagiert klar und sicher.

Praxisbeispiel: Artikel veraltet nach Release

Situation

Nach einem Release ändert sich der Menüpfad für eine Funktion.

Problem

Der alte Benutzerartikel beschreibt den früheren Menüpfad.

Folge

Benutzer erstellen Tickets, obwohl die Funktion verfügbar ist.

Verbesserung

Change- und Release-Abschluss enthalten künftig eine Prüfung betroffener Knowledge-Artikel.

Typische Fehler

Fehler 1

Known Error ist bekannt, aber nicht dokumentiert.

Fehler 2

Workaround steht nur in einem alten Ticket.

Fehler 3

Service Desk kennt den Workaround nicht.

Fehler 4

Artikel beschreibt nicht, wann der Workaround nicht gilt.

Fehler 5

Risiken werden nicht genannt.

Fehler 6

Known Error wird nicht mit Problem Record verknüpft.

Fehler 7

Artikel wird nach Change oder Release nicht aktualisiert.

Fehler 8

Benutzer erhalten interne technische Details.

Fehler 9

Kein Workaround verfügbar, aber der Artikel sagt das nicht ausdrücklich.

Fehler 10

Service Desk dokumentiert nicht, welcher Artikel genutzt wurde.

Fehler 11

Häufig genutzte Workarounds werden nicht an Problem Management gemeldet.

Fehler 12

Feedback aus dem Service Desk wird nicht bearbeitet.

Checkliste Known-Error-Artikel

- ☐ Fehlerbild klar beschrieben
 - ☐ betroffener Service genannt
 - ☐ betroffene Version oder Komponente genannt
 - ☐ Ursache oder wahrscheinliche Ursache beschrieben
 - ☐ Workaround vorhanden oder ausdrücklich nicht verfügbar
 - ☐ Risiken genannt
 - ☐ Einschränkungen genannt
 - ☐ Prüfschritte enthalten
 - ☐ Eskalationskriterien beschrieben
 - ☐ Problem Record verknüpft
 - ☐ geplanter Change oder dauerhafte Lösung genannt
 - ☐ Owner und Review-Datum vorhanden
-

Checkliste Workaround

- ☐ genaue Schritte beschrieben
- ☐ Voraussetzungen genannt
- ☐ Zielgruppe klar
- ☐ erwartetes Ergebnis beschrieben

- ☐ Risiken bewertet
 - ☐ Abbruchkriterien definiert
 - ☐ Eskalationsweg genannt
 - ☐ Dokumentation im Ticket beschrieben
 - ☐ Gültigkeitsbereich klar
 - ☐ Review nach Change oder Release vorgesehen
-

Checkliste Service Desk Nutzung

- ☐ bei bekannten Symptomen Knowledge Base prüfen
 - ☐ passenden Artikel im Ticket verknüpfen
 - ☐ Prüfschritte dokumentieren
 - ☐ Workaround korrekt anwenden
 - ☐ Ergebnis dokumentieren
 - ☐ bei Abweichung eskalieren
 - ☐ bei falschem Artikel Feedback geben
 - ☐ wiederkehrende Fälle an Problem Management melden
 - ☐ Benutzer mit freigegebenem Text informieren
 - ☐ Artikel nicht blind anwenden, wenn Fehlerbild nicht passt
-

Checkliste Wissenspflege

- ☐ häufig genutzte Workarounds auswerten
 - ☐ Known Errors regelmäßig prüfen
 - ☐ Artikel nach Changes aktualisieren
 - ☐ Artikel nach Releases aktualisieren
 - ☐ gelöste Known Errors archivieren oder anpassen
 - ☐ Feedback des Service Desk bearbeiten
 - ☐ Suchbegriffe verbessern
 - ☐ doppelte Artikel zusammenführen
 - ☐ veraltete Workarounds entfernen
 - ☐ Problem Management über neue Muster informieren
-

Bedeutung für Fachinformatiker für Systemintegration

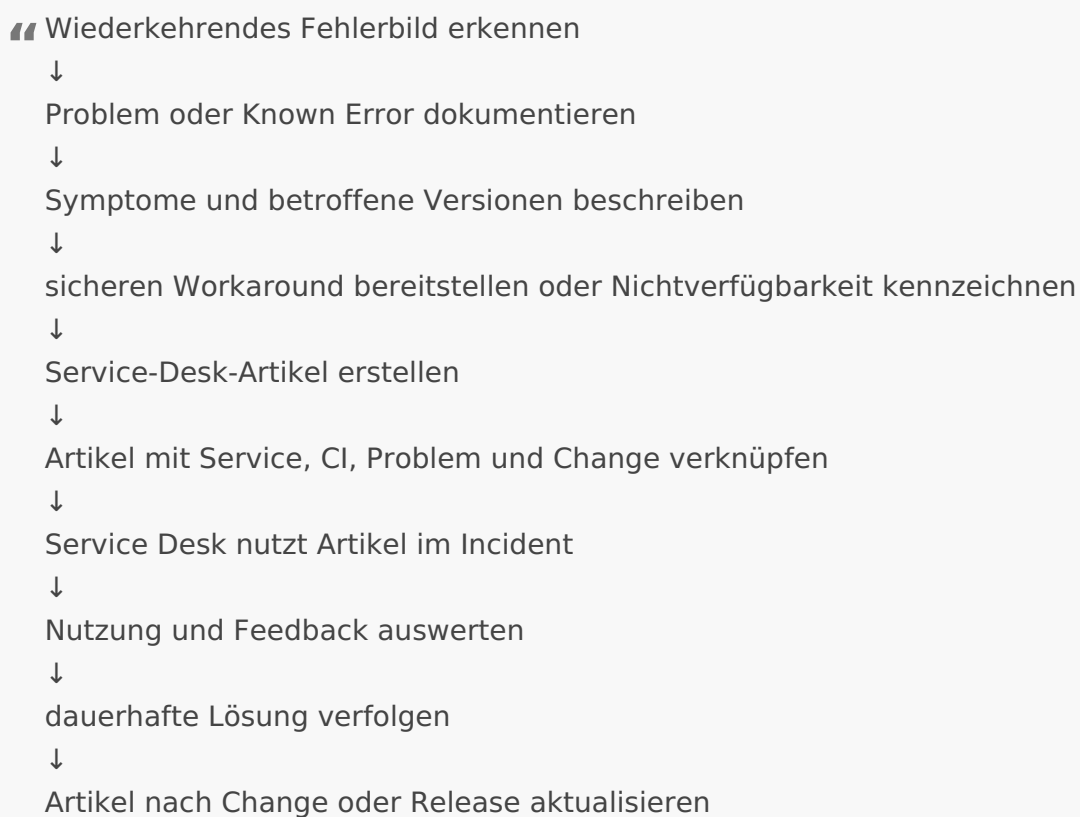
Fachinformatiker unterstützen Known Errors und Workarounds durch technisches Verständnis und saubere Dokumentation.

Im Arbeitsalltag bedeutet das:

- wiederkehrende Fehlerbilder erkennen,
- technische Prüfschritte verständlich beschreiben,
- Workarounds sicher dokumentieren,
- Risiken und Grenzen nennen,
- Known Errors mit CIs und Problems verknüpfen,
- Service Desk mit klaren Artikeln unterstützen,
- nach Changes Artikel aktualisieren,
- und häufig genutzte Workarounds als Hinweis auf ungelöste Problems verstehen.

Gute Workaround- und Known-Error-Dokumentation spart Zeit und erhöht die Qualität im Support.

Zusammenfassung



Merksätze

Ein Known Error muss auffindbar sein, sonst hilft er dem Service Desk nicht.

“ Ein Workaround ist hilfreich, aber keine automatische Dauerlösung.

“ Jeder Workaround braucht Grenzen, Risiken und Eskalationskriterien.

“ Kein Workaround verfügbar ist ebenfalls eine wichtige Information.

“ Service Desk Feedback verbessert die Knowledge Base.

“ Häufig genutzte Workarounds sind ein Hinweis auf ungelöste Problems.

“ Nach Changes und Releases müssen Known Errors und Workarounds überprüft werden.

Verwandte Seiten

- 7.1 Knowledge Management – Ziele, Begriffe und Grundlagen
- 7.2 Knowledge-Artikel, Runbooks und FAQ
- 7.4 Self-Service und Benutzerwissen
- 7.5 Knowledge Management im Zusammenspiel mit Incident, Problem und Change
- 4.3 Workarounds und Known Errors
- Incident Management
- Problem Management
- Change Enablement
- Service Configuration Management

Quellen und Versionsstand

Offizielle Grundlagen

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

Ergänzende Praxiseinordnung

- Knowledge-Centered Service als verbreiteter Praxisansatz für Wissensarbeit im Support

Einordnung

Die dargestellten:

- Known-Error-Strukturen,
- Workaround-Hinweise,
- Service-Desk-Checklisten,
- Ticketdokumentation,
- Pflegehinweise,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Known-Error-Vorlage,
- Workaround-Struktur,
- Service-Desk-Artikelpflicht,
- Eskalationsmatrix,
- Review-Frequenz,
- oder Toolintegration

für alle Organisationen vor.

Die konkrete Umsetzung muss an:

- Services,
- Risiken,
- Supportmodell,
- Knowledge-Werkzeuge,
- Problem Management,
- Change-Modell,
- Sicherheitsanforderungen,
- Service Configuration Management,
- und verfügbare Fähigkeiten

angepasst werden.

Behandelter Framework-Stand: ITIL Version 5

Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance

Fachlicher Stand: August 2026
