8.4 Service Reviews und Continual Improvement

> **Kurz erklärt**
>
> Service Reviews bewerten regelmäßig, wie gut ein IT-Service seine vereinbarten Ziele erfüllt und ob er den erwarteten Nutzen liefert.
>
> Dabei werden nicht nur SLA-Werte betrachtet.
>
> Auch Incidents, Problems, Changes, Benutzerfeedback, Risiken, Lieferantenleistungen und Verbesserungspotenziale gehören dazu.
>
> Continual Improvement sorgt anschließend dafür, dass aus Erkenntnissen konkrete und überprüfbare Verbesserungen entstehen.

Warum Service Reviews wichtig sind

Ein Report allein verbessert keinen Service.

Kennzahlen zeigen beispielsweise:

- Verfügbarkeit,
- Reaktionszeiten,
- Wiederherstellungszeiten,
- Ticketvolumen,
- SLA-Erfüllung,
- Benutzerzufriedenheit,
- Changes,
- Major Incidents.

Ein Service Review bewertet zusätzlich:

- Warum wurde ein Ziel erreicht oder verfehlt?
- Welche Auswirkungen hatten Abweichungen?
- Welche Risiken bestehen?
- Welche Ursachen sind bekannt?
- Welche Maßnahmen laufen bereits?
- Welche Entscheidung wird benötigt?
- Welche Verbesserung sollte priorisiert werden?

Ein Service Review verbindet Daten mit Bewertung, Verantwortung und Handlung.

Was ist ein Service Review?

Ein Service Review ist eine regelmäßige strukturierte Betrachtung eines Service.

Dabei wird gemeinsam geprüft:

- aktuelle Servicequalität,
- Zielerreichung,
- Benutzererfahrung,
- wichtige Incidents,
- offene Problems,
- relevante Changes,
- Lieferantenleistungen,
- bekannte Risiken,
- geplante Entwicklungen,
- und Verbesserungsmaßnahmen.

Service Reviews können je nach Kritikalität und Organisation:

- monatlich,
- quartalsweise,
- halbjährlich,
- oder anlassbezogen

stattfinden.

Ziele eines Service Reviews

Ein Service Review soll:

- Servicequalität transparent machen,
- Abweichungen verstehen,
- Risiken frühzeitig erkennen,
- Verantwortlichkeiten klären,
- Maßnahmen priorisieren,
- Entscheidungen vorbereiten,
- Kunden- und Benutzerfeedback einbeziehen,
- Lieferantenleistung bewerten,
- und Continual Improvement unterstützen.

Das Review sollte nicht nur rückwärts schauen.

Es sollte auch kommende Risiken und Veränderungen betrachten.

Service Review ist keine reine Zahlenbesprechung

Ein häufiger Fehler:

> Im Service Review werden nur Diagramme und SLA-Werte vorgelesen.

Das reicht nicht aus.

Ein gutes Service Review beantwortet:

- Was bedeutet die Zahl?
- Warum ist sie so?
- Welche Benutzer waren betroffen?
- Ist ein Trend erkennbar?
- Muss ein Problem untersucht werden?
- Muss ein Change geplant werden?
- Muss ein Lieferant eskaliert werden?
- Muss ein Service Level angepasst werden?
- Welche Maßnahme wird bis wann umgesetzt?

> **Merke**
>
> Reporting liefert Informationen.
>
> Das Service Review bewertet diese Informationen und löst Entscheidungen aus.

Teilnehmer eines Service Reviews

Mögliche Teilnehmer:

- Service Owner,
- Kunde oder Fachbereich,
- Service Level Manager,
- Service Desk,
- technische Supportgruppen,
- Problem Management,
- Change Management,
- Supplier Management,
- Information Security,
- Continual Improvement,
- Management,
- wichtige Lieferanten.

Nicht jedes Review benötigt alle Rollen.

Die Teilnehmer sollten zum Service und zu den aktuellen Themen passen.

Rolle des Service Owners

Der Service Owner trägt eine zentrale Verantwortung.

Typische Aufgaben:

- Serviceleistung bewerten,
- Risiken vertreten,
- Stakeholder einbinden,
- Verbesserungen priorisieren,
- Entscheidungen vorbereiten,
- Maßnahmen verfolgen,
- Abhängigkeiten berücksichtigen,
- und Serviceziele aktuell halten.

Der Service Owner muss nicht jede technische Einzelheit kennen.

Er muss aber verstehen, welche Auswirkungen technische Themen auf den Service haben.

Rolle des Service Level Managers

Der Service Level Manager unterstützt durch:

- Vorbereitung des Reviews,
- Zusammenstellung von Kennzahlen,
- Prüfung der SLA-Erfüllung,
- Abstimmung mit Kunden und Teams,
- Dokumentation von Abweichungen,
- Nachverfolgung von Maßnahmen,
- und Pflege der Service-Level-Vereinbarungen.

Die Rolle kann je nach Organisation anders benannt oder verteilt sein.

Rolle des Service Desk

Der Service Desk liefert wichtige Praxiserkenntnisse.

Beispiele:

- häufige Benutzerfragen,
- wiederkehrende Incidents,
- unklare Self-Service-Inhalte,
- Probleme bei Eskalationen,

- schlechte Knowledge-Artikel,
- Benutzerfeedback,
- häufige Workarounds,
- auffällige Tickettrends.

Der Service Desk kennt oft die tatsächliche Benutzerwahrnehmung besser als rein technische Reports.

Rolle der technischen Teams

Technische Teams liefern Informationen zu:

- Ursachen,
- Plattformzustand,
- Kapazität,
- Monitoring,
- technischen Risiken,
- Changes,
- Releases,
- Sicherheitslücken,
- Abhängigkeiten,
- und geplanten Maßnahmen.

Technische Informationen sollten so erklärt werden, dass ihre Servicewirkung verständlich ist.

Rolle des Fachbereichs oder Kunden

Fachbereiche und Kunden liefern die Geschäftsperspektive.

Wichtige Fragen:

- Unterstützt der Service den Arbeitsprozess?
- Welche Funktionen sind besonders wichtig?
- Welche Störungen waren besonders kritisch?
- Haben sich Anforderungen verändert?
- Gibt es neue Benutzergruppen?
- Sind Support und Kommunikation ausreichend?
- Welche Verbesserungen haben den höchsten Nutzen?

Servicequalität darf nicht nur aus IT-Sicht bewertet werden.

Vorbereitung eines Service Reviews

Vor dem Review sollten relevante Informationen gesammelt werden.

Mögliche Inhalte:

- SLA- und Service-Level-Berichte,
- Verfügbarkeitsdaten,
- Incident-Trends,
- Major Incidents,
- Problem Records,
- Known Errors,
- häufig genutzte Workarounds,
- Change-Ergebnisse,
- Failed Changes,
- Release-Ergebnisse,
- Benutzerfeedback,
- Lieferantenberichte,
- Security-Risiken,
- Kapazitätsdaten,
- offene Verbesserungsmaßnahmen.

Die Vorbereitung sollte sich auf wichtige Themen konzentrieren.

Ein unübersichtlicher Datenberg erschwert Entscheidungen.

Mögliche Agenda

Eine kompakte Agenda kann so aussehen:

1. Servicezustand und wichtige Entwicklungen
2. Zielerreichung und SLA-Abweichungen
3. Benutzer- und Kundenfeedback
4. wichtige Incidents und Major Incidents
5. offene Problems und Known Errors
6. Changes und Releases
7. Lieferantenleistung
8. Risiken und kommende Anforderungen
9. Verbesserungsmaßnahmen
10. Entscheidungen, Owner und Termine

Die Agenda sollte an den Service angepasst werden.

Servicezustand bewerten

Zu Beginn sollte ein Gesamtbild entstehen.

Mögliche Fragen:

- Ist der Service stabil?
- Gibt es aktuell kritische Risiken?
- Haben sich Nutzerzahlen verändert?
- Hat sich die Nutzung verändert?
- Gibt es technische Engpässe?
- Gibt es bevorstehende Änderungen?
- Sind Lieferantenleistungen stabil?
- Gibt es neue Sicherheitsanforderungen?

Ein Gesamtbild hilft, einzelne Kennzahlen richtig einzuordnen.

SLA-Erfüllung bewerten

Zu prüfen ist:

- Welche Ziele wurden erreicht?
- Welche Ziele wurden verfehlt?
- Wie groß war die Abweichung?
- Handelt es sich um einen Einzelfall oder Trend?
- Welche Benutzer waren betroffen?
- Welche Ursache ist bekannt?
- Welche Gegenmaßnahmen laufen?
- Muss ein Problem Record erstellt werden?
- Muss das SLA angepasst werden?

Eine Zielverfehlung muss nicht automatisch bedeuten, dass das Ziel falsch ist.

Aber sie muss verstanden werden.

Erreichte Ziele kritisch betrachten

Auch grüne Kennzahlen sollten hinterfragt werden.

Beispiele:

- Wurde das Ziel erreicht, weil es zu niedrig ist?
- Wurde eine automatische Antwort als Reaktion gezählt?
- Wurden Wartungszeiten großzügig ausgeschlossen?
- Sind Benutzer trotzdem unzufrieden?
- Werden kritische Einzelfälle durch Durchschnittswerte verdeckt?
- Stimmen Daten und Servicewahrnehmung überein?

Ein grünes Dashboard ist kein Beweis für einen guten Service.

Benutzerfeedback einbeziehen

Benutzerfeedback kann stammen aus:

- Ticketumfragen,
- Serviceportal,
- Interviews,
- Beschwerden,
- Fachbereichsgesprächen,
- Service-Desk-Rückmeldungen,
- Artikelbewertungen,
- Workshops,
- oder Nutzungsanalysen.

Zu prüfen ist:

- Welche Themen wiederholen sich?
- Welche Probleme sind technisch nicht sichtbar?
- Welche Prozesse sind zu kompliziert?
- Welche Kommunikation ist unverständlich?
- Welche Self-Service-Angebote fehlen?
- Welche Funktionen sind besonders wichtig?

Qualitatives Feedback ergänzt Kennzahlen.

Incidents betrachten

Im Service Review sollten nicht alle Incidents einzeln besprochen werden.

Wichtig sind:

- Major Incidents,
- wiederkehrende Incidents,
- kritische Incidents,
- Incidents mit langer Wiederherstellung,
- Incidents nach Changes,
- Incidents mit schlechter Kommunikation,
- häufige Eskalationen,
- auffällige Standorte oder Benutzergruppen.

Ziel ist, Muster und Verbesserungsmöglichkeiten zu erkennen.

Major Incidents betrachten

Bei Major Incidents sollte geprüft werden:

- Was war die Ursache?

- Wie schnell wurde reagiert?
- War Ownership klar?
- War Kommunikation ausreichend?
- Funktionierten Eskalationswege?
- Waren Runbooks aktuell?
- Gab es einen Workaround?
- Wurde ein Problem Record erstellt?
- Wurden Lessons Learned umgesetzt?
- Sind weitere Maßnahmen offen?

Ein Major-Incident-Review kann Teil eines Service Reviews sein oder separat stattfinden.

Problems und Known Errors betrachten

Wichtige Fragen:

- Welche offenen Problems betreffen den Service?
- Welche Known Errors bestehen?
- Wie häufig werden Workarounds genutzt?
- Gibt es eine dauerhafte Lösung?
- Ist ein Change geplant?
- Ist das Risiko noch akzeptabel?
- Sind Service Desk und Benutzer informiert?
- Müssen Prioritäten angepasst werden?
- Gibt es gealterte Problem Records?

Häufig genutzte Workarounds sind ein Warnsignal.

Changes betrachten

Zu prüfen ist:

- Welche wichtigen Changes wurden umgesetzt?
- Haben sie den erwarteten Nutzen erreicht?
- Gab es Failed Changes?
- Sind Incidents nach Changes entstanden?
- Wurde Knowledge aktualisiert?
- Wurde die CMDB aktualisiert?
- Wurden Risiken korrekt bewertet?
- Stehen wichtige Changes bevor?
- Gibt es Change-Kollisionen oder Freeze-Zeiten?

Changes beeinflussen Servicequalität direkt.

Releases betrachten

Bei Releases sind wichtig:

- erfolgreiche Einführung,
- Benutzerfeedback,
- Incidents nach Release,
- bekannte Fehler,
- neue Workarounds,
- Supportaufwand,
- aktualisierte Dokumentation,
- aktualisierte Knowledge-Artikel,
- und erreichte Geschäftsziele.

Ein technisch erfolgreiches Deployment kann aus Benutzersicht trotzdem unzureichend sein.

Lieferantenleistung betrachten

Bei externen Abhängigkeiten sollte geprüft werden:

- Wurden Vertragsziele erreicht?
- Waren Reaktionszeiten ausreichend?
- Gab es wiederkehrende Störungen?
- Waren Eskalationswege wirksam?
- War die Kommunikation ausreichend?
- Gibt es offene Herstellerfehler?
- Müssen Verträge angepasst werden?
- Bestehen Abhängigkeiten von einzelnen Lieferanten?
- Gibt es Alternativen oder Notfallverfahren?

Lieferantenprobleme dürfen nicht einfach als unveränderlich hingenommen werden.

Risiken betrachten

Mögliche Servicerisiken:

- veraltete Software,
- ablaufende Zertifikate,
- fehlende Redundanz,
- bekannte Kapazitätsgrenzen,
- unklare Owner,
- fehlende Dokumentation,
- häufige Emergency Changes,
- unzureichende Lieferantenleistung,
- fehlende Backuptests,

- Sicherheitslücken,
- hohe Abhängigkeit von Einzelpersonen,
- geplanter Hersteller-Supportablauf.

Risiken sollten mit Auswirkung, Wahrscheinlichkeit, Owner und Maßnahme dokumentiert werden.

Kommende Anforderungen betrachten

Ein Service Review sollte auch nach vorne schauen.

Beispiele:

- neue Benutzergruppen,
- steigende Nutzerzahlen,
- neue Standorte,
- neue gesetzliche Anforderungen,
- Cloud-Migration,
- neues Release,
- Vertragsende,
- End of Support,
- neue Sicherheitsanforderungen,
- geänderte Geschäftszeiten,
- neue Schnittstellen,
- geplante Umstrukturierung.

So können notwendige Changes und Kapazitäten frühzeitig geplant werden.

Continual Improvement

Continual Improvement bedeutet, Services, Practices, Prozesse, Werkzeuge und Arbeitsweisen fortlaufend zu verbessern.

Verbesserungen können klein oder groß sein.

Beispiele:

- Knowledge-Artikel korrigieren,
- Monitoringalarm verbessern,
- Formular vereinfachen,
- OLA-Eskalation anpassen,
- Zertifikatsüberwachung einführen,
- Standard Change definieren,
- Runbook aktualisieren,
- Lieferantenvertrag verbessern,
- Kapazität erweitern,

- Self-Service automatisieren.

Nicht jede Verbesserung benötigt ein großes Projekt.

Continual Improvement ist keine einmalige Maßnahme

Ein häufiger Irrtum:

> ❝ Nach einem Verbesserungsprojekt ist der Service dauerhaft optimiert.

Services verändern sich ständig.

Gründe:

- neue Anforderungen,
- neue Technologien,
- neue Benutzer,
- neue Risiken,
- neue Lieferanten,
- neue Sicherheitslage,
- neue Fehler,
- neue gesetzliche Vorgaben.

Deshalb ist Continual Improvement ein fortlaufender Kreislauf.

Verbesserungskreislauf

Ein möglicher Ablauf:

```
aktuelle Situation verstehen
        ↓
Zielzustand festlegen
        ↓
Verbesserungspotenzial erkennen
        ↓
Maßnahme priorisieren
        ↓
Umsetzung planen
        ↓
Verbesserung durchführen
```

```
        ↓
Wirkung messen
        ↓
Ergebnis bewerten
        ↓
nächste Verbesserung auswählen
```

Wichtig ist, nach der Umsetzung zu prüfen, ob die gewünschte Wirkung tatsächlich erreicht wurde.

Improvement Register

Ein Improvement Register sammelt Verbesserungsideen und Maßnahmen.

Mögliche Felder:

- Titel,
- Beschreibung,
- Auslöser,
- betroffener Service,
- erwarteter Nutzen,
- Risiko,
- Priorität,
- Aufwand,
- Owner,
- Zieltermin,
- Status,
- Abhängigkeiten,
- Erfolgskriterien,
- Ergebnis.

Das Improvement Register kann in einem ITSM-System, Projekttool, Wiki oder einer strukturierten Liste geführt werden.

Warum ein Improvement Register wichtig ist

Ohne zentrale Nachverfolgung entstehen typische Probleme:

- gute Ideen gehen verloren,
- Maßnahmen werden nicht zugewiesen,
- gleiche Verbesserung wird mehrfach vorgeschlagen,
- niemand kennt den Status,
- dringende Themen verdrängen wichtige langfristige Maßnahmen,
- Lessons Learned werden nicht umgesetzt.

Das Register schafft Transparenz und Verantwortung.

Verbesserungsideen erfassen

Verbesserungsideen können entstehen aus:

- Service Reviews,
- Major Incidents,
- Problem Management,
- Failed Changes,
- Benutzerfeedback,
- Service Desk,
- Audits,
- Security Reviews,
- Lieferantenreviews,
- Datenqualitätsberichten,
- Knowledge-Auswertungen,
- Monitoringtrends,
- Mitarbeitervorschlägen.

Jede Idee sollte kurz beschreiben:

- Was ist das Problem?
- Welche Auswirkung besteht?
- Welche Verbesserung wird vorgeschlagen?
- Welcher Nutzen wird erwartet?

Verbesserungen priorisieren

Nicht alle Verbesserungen können gleichzeitig umgesetzt werden.

Mögliche Kriterien:

- geschäftlicher Nutzen,
- Risiko,
- Anzahl betroffener Benutzer,
- Kritikalität des Services,
- Sicherheitsrelevanz,
- Häufigkeit des Problems,
- Kosten,
- Aufwand,
- Dringlichkeit,
- Abhängigkeiten,
- gesetzliche Anforderungen,
- erwartete Wirkung.

Eine einfache Maßnahme mit hohem Nutzen kann vor einem großen Projekt sinnvoll sein.

Nutzen und Aufwand vergleichen

Eine einfache Bewertung kann helfen:

Nutzen	Aufwand	mögliche Priorität
hoch	niedrig	sehr hoch
hoch	hoch	strategisch bewerten
niedrig	niedrig	bei Gelegenheit
niedrig	hoch	meist geringe Priorität

Diese Matrix ist nur ein Beispiel.

Sicherheits- oder Compliance-Themen können unabhängig vom Aufwand hohe Priorität besitzen.

Erfolgskriterien für Verbesserungen

Vor Umsetzung sollte klar sein, woran Erfolg erkannt wird.

Beispiele:

Verbesserung	Erfolgskriterium
Knowledge-Artikel verbessern	weniger Tickets und bessere Bewertungen
Monitoring erweitern	Fehler wird vor Benutzerbeschwerden erkannt
OLA anpassen	Eskalationszeit sinkt
Self-Service einführen	Standardrequests werden schneller erfüllt
Zertifikatsmonitoring einführen	keine ungeplanten Ausfälle durch Ablauf
Change-Checkliste ergänzen	weniger Incidents nach Changes

Ohne Erfolgskriterien bleibt unklar, ob die Verbesserung wirksam war.

Kleine Verbesserungen

Kleine Verbesserungen können schnell Wirkung zeigen.

Beispiele:

- unklaren Artikeltitel ändern,
- fehlenden Owner ergänzen,

- Ticketformular vereinfachen,
- Eskalationskontakt aktualisieren,
- Monitoringgrenze korrigieren,
- häufigen Workaround dokumentieren,
- Service-Desk-Vorlage verbessern.

Solche Verbesserungen sollten nicht unnötig durch große Prozesse verzögert werden.

Größere Verbesserungen

Größere Maßnahmen können sein:

- Hochverfügbarkeit einführen,
- Plattform migrieren,
- neuen Service Desk einführen,
- Lieferanten wechseln,
- Self-Service-Portal aufbauen,
- CMDB einführen,
- Monitoring grundlegend modernisieren,
- Service neu gestalten.

Solche Maßnahmen benötigen oft:

- Business Case,
- Projekt,
- Changes,
- Risikoanalyse,
- Budget,
- Ressourcen,
- Kommunikation,
- und Erfolgsüberwachung.

Verbesserung und Change Enablement

Viele Verbesserungen werden durch Changes umgesetzt.

Beispiele:

- neue Monitoringregel,
- Konfigurationsänderung,
- Automatisierung,
- Softwareupdate,
- neuer Workflow,
- neues Formular,
- neue Plattform.

Continual Improvement erkennt und priorisiert den Verbesserungsbedarf.

Change Enablement sorgt für kontrollierte Umsetzung.

Verbesserung und Problem Management

Problem Management liefert Ursachen und dauerhafte Lösungsvorschläge.

Beispiele:

- wiederkehrende Incidents,
- häufig genutzte Workarounds,
- technische Schwachstellen,
- unklare Abhängigkeiten,
- fehlerhafte Versionen.

Diese Erkenntnisse können als Improvement erfasst und priorisiert werden.

Verbesserung und Knowledge Management

Knowledge Management kann verbessert werden durch:

- neue Artikel,
- aktualisierte Runbooks,
- bessere Suchbegriffe,
- klarere Zielgruppen,
- neue FAQ,
- archivierte Altartikel,
- bessere Verknüpfungen,
- Review-Prozesse,
- Feedbackauswertung.

Wissen ist häufig eine schnelle und kostengünstige Verbesserungsmöglichkeit.

Verbesserung und Service Configuration Management

Configuration-Daten können verbessert werden durch:

- fehlende CIs ergänzen,
- Owner aktualisieren,
- Beziehungen pflegen,
- Zertifikate erfassen,
- Cloud-Tags einführen,
- Dubletten bereinigen,
- Change-Prozess anbinden,

- Service Maps erstellen.

Gute Configuration-Daten verbessern Incident, Problem, Change und Security Management.

Verbesserung und Supplier Management

Lieferantenverbesserungen können sein:

- Eskalationsweg klären,
- Supportzeiten anpassen,
- Reporting verbessern,
- Vertragsziel ändern,
- Lieferantenreview einführen,
- technische Alternative prüfen,
- zusätzliche Redundanz schaffen,
- Supportkompetenz intern aufbauen.

Lieferantenleistung ist Teil der gesamten Servicequalität.

Verbesserungen verfolgen

Jede wichtige Maßnahme sollte mindestens besitzen:

- klare Beschreibung,
- Owner,
- Priorität,
- Zieltermin,
- Status,
- erwarteten Nutzen,
- Erfolgskriterium.

Ungeeignet:

> Monitoring verbessern.

Besser:

> Bis zum 30.09.2026 Zertifikatsmonitoring für alle produktiven Webservices einführen. Erfolg: Warnung mindestens 30 Tage vor Ablauf, Owner je Zertifikat dokumentiert.

Status von Verbesserungen

Mögliche Status:

- vorgeschlagen,
- wird bewertet,
- genehmigt,
- geplant,
- in Umsetzung,
- blockiert,
- umgesetzt,
- Wirkung wird geprüft,
- abgeschlossen,
- verworfen.

Ein klarer Status erleichtert Nachverfolgung.

Blockierte Verbesserungen

Verbesserungen können blockiert sein durch:

- fehlendes Budget,
- fehlende Ressourcen,
- Lieferantenabhängigkeit,
- technische Abhängigkeit,
- fehlende Entscheidung,
- Change Freeze,
- Sicherheitsprüfung,
- unklare Verantwortung.

Blockierungen sollten sichtbar sein.

Zu dokumentieren ist:

- Ursache,
- verantwortliche Stelle,
- benötigte Entscheidung,
- nächste Prüfung,
- Risiko der Verzögerung.

Wirkung messen

Nach Umsetzung muss geprüft werden:

- Hat sich die Kennzahl verbessert?

- Ist das ursprüngliche Problem reduziert?
- Sind Benutzer zufriedener?
- Gibt es weniger Incidents?
- Wurde das Risiko reduziert?
- Funktioniert der neue Prozess?
- Sind neue Nebenwirkungen entstanden?
- Muss nachgebessert werden?

Eine umgesetzte Maßnahme ist nicht automatisch eine erfolgreiche Verbesserung.

Verbesserung schließen

Eine Verbesserung sollte erst geschlossen werden, wenn:

- Maßnahme umgesetzt wurde,
- Ergebnis geprüft wurde,
- Erfolgskriterien bewertet wurden,
- Dokumentation aktualisiert wurde,
- Rest Risiko bekannt ist,
- Stakeholder informiert wurden,
- und gegebenenfalls weitere Maßnahmen erfasst wurden.

Wenn die Wirkung noch nicht messbar ist, kann der Status „Wirkung wird geprüft“ sinnvoll sein.

Service Review dokumentieren

Ein Service Review sollte dokumentieren:

- Datum,
- Teilnehmer,
- betrachteter Zeitraum,
- wichtigste Kennzahlen,
- Abweichungen,
- Risiken,
- Entscheidungen,
- Maßnahmen,
- Owner,
- Termine,
- offene Fragen,
- nächster Review-Termin.

Die Dokumentation sollte kompakt und handlungsorientiert sein.

Entscheidungen dokumentieren

Wichtige Entscheidungen können sein:

- SLA-Ziel anpassen,
- Problem Record priorisieren,
- Change starten,
- Lieferant eskalieren,
- Risiko akzeptieren,
- zusätzliche Kapazität bereitstellen,
- Serviceumfang ändern,
- Knowledge verbessern,
- Monitoring erweitern.

Zu jeder Entscheidung sollte nachvollziehbar sein:

- wer entschieden hat,
- warum entschieden wurde,
- welche Auswirkung erwartet wird,
- und wann überprüft wird.

Maßnahmenprotokoll

Ein einfaches Maßnahmenprotokoll:

Maßnahme	Owner	Termin	Status	Erfolgskriterium
VPN-Known-Error aktualisieren	Netzwerkteam	15.08.2026	offen	Service Desk findet Artikel
Zertifikatsmonitoring einführen	Plattformteam	30.09.2026	geplant	Warnung 30 Tage vor Ablauf
OLA-Eskalation anpassen	Service Level Manager	31.08.2026	in Arbeit	kürzere P1-Eskalationszeit

Ein Maßnahmenprotokoll verhindert, dass Entscheidungen nach dem Review verloren gehen.

Folgereview

Im nächsten Service Review sollte geprüft werden:

- Wurden Maßnahmen umgesetzt?
- Wurden Termine eingehalten?
- Welche Wirkung ist erkennbar?
- Welche Blockierungen bestehen?
- Muss eine Maßnahme angepasst werden?
- Sind neue Risiken entstanden?
- Kann eine Verbesserung geschlossen werden?

Ohne Folgereview verlieren Maßnahmen schnell an Verbindlichkeit.

Praxisbeispiel: Wiederkehrende VPN-Incidents

Beobachtung

VPN-Incidents steigen über drei Monate.

Service Review

- betroffene Clientversion identifiziert,
- Workaround wird häufig genutzt,
- Benutzerfeedback negativ,
- Known Error vorhanden,
- dauerhafte Lösung noch nicht umgesetzt.

Verbesserung

- neue Clientversion pilotieren,
- Rollout planen,
- Knowledge-Artikel aktualisieren,
- Incident-Trend nach Rollout prüfen.

Erfolgskriterium

VPN-Incidents sinken innerhalb von vier Wochen deutlich.

Praxisbeispiel: SLA erfüllt, Benutzer unzufrieden

Beobachtung

Verfügbarkeitsziel wurde erreicht.

Benutzer melden trotzdem schlechte Performance.

Service Review

- Verfügbarkeit misst nur Erreichbarkeit,
- Antwortzeiten werden nicht betrachtet,
- Fachbereich meldet lange Ladezeiten.

Verbesserung

- Performance-Kennzahl ergänzen,
- synthetische Transaktion einführen,
- Kapazitätsanalyse starten,

- Benutzerfeedback im nächsten Review prüfen.

Praxisbeispiel: Lieferantenproblem

Beobachtung

Mehrere SLA-Abweichungen entstehen durch langsame Herstellerreaktion.

Service Review

- Underpinning Contract passt nicht zum internen SLA,
- Eskalationsweg ist unklar,
- kein technischer Workaround vorhanden.

Verbesserung

- Lieferantenvertrag prüfen,
- Eskalationsweg definieren,
- interne technische Kompetenz ausbauen,
- SLA-Risiko transparent machen.

Praxisbeispiel: Major Incident

Review-Erkenntnisse

- Monitoring erkannte Fehler zu spät,
- Service Desk hatte kein aktuelles Runbook,
- Statuskommunikation war uneinheitlich,
- Owner war unklar.

Verbesserungen

- Monitoring erweitern,
- Runbook aktualisieren,
- Kommunikationsvorlage erstellen,
- Ownership im Major-Incident-Prozess festlegen,
- Wirksamkeit bei nächster Übung prüfen.

Praxisbeispiel: Self-Service verbessern

Beobachtung

Viele Tickets entstehen zur MFA-Einrichtung.

Analyse

- Artikel schwer auffindbar,
- Suchbegriffe fehlen,
- Formular ist kompliziert,
- Benutzer besitzen neues Smartphone.

Verbesserungen

- neuen Artikel erstellen,
- Suchbegriffe ergänzen,
- Formular vereinfachen,
- Self-Service-Link im Onboarding ergänzen.

Erfolgskriterium

Weniger MFA-Standardtickets und bessere Artikelbewertung.

Typische Fehler

Fehler 1

Service Reviews bestehen nur aus dem Vorlesen von Kennzahlen.

Fehler 2

Es nehmen zu viele Personen ohne klare Rolle teil.

Fehler 3

Benutzer- und Fachbereichsfeedback fehlt.

Fehler 4

Grüne Kennzahlen werden nicht kritisch hinterfragt.

Fehler 5

SLA-Abweichungen werden erklärt, aber keine Maßnahmen beschlossen.

Fehler 6

Maßnahmen besitzen keinen Owner.

Fehler 7

Verbesserungen besitzen kein Erfolgskriterium.

Fehler 8

Lessons Learned werden nicht verfolgt.

Fehler 9

Lieferantenleistung wird getrennt vom Service betrachtet.

Fehler 10

Reviews schauen nur zurück und nicht auf kommende Risiken.

Fehler 11

Umgesetzte Maßnahmen werden nicht auf Wirkung geprüft.

Fehler 12

Verbesserungsideen gehen außerhalb eines Registers verloren.

Checkliste Service Review vorbereiten

- [ ] betrachteter Service ist klar
- [ ] Berichtszeitraum festgelegt
- [ ] SLA- und Kennzahlendaten vorhanden
- [ ] Datenqualität geprüft
- [ ] wichtige Incidents ausgewählt
- [ ] Major Incidents berücksichtigt
- [ ] offene Problems und Known Errors geprüft
- [ ] Changes und Releases berücksichtigt
- [ ] Benutzerfeedback gesammelt
- [ ] Lieferantenleistung geprüft
- [ ] Risiken aktualisiert
- [ ] offene Maßnahmen aus letztem Review geprüft

- [ ] passende Teilnehmer eingeladen

Checkliste Service Review durchführen

- [ ] Gesamtzustand des Services bewertet
- [ ] Zielerreichung geprüft
- [ ] SLA-Abweichungen analysiert
- [ ] Benutzererfahrung berücksichtigt
- [ ] Incident-Trends betrachtet
- [ ] Problems und Workarounds geprüft
- [ ] Change- und Release-Wirkung bewertet
- [ ] Lieferantenleistung betrachtet
- [ ] kommende Risiken besprochen
- [ ] Verbesserungen priorisiert
- [ ] Entscheidungen dokumentiert
- [ ] Owner und Termine festgelegt

Checkliste Improvement Register

- [ ] Verbesserung verständlich beschrieben
- [ ] Auslöser dokumentiert
- [ ] betroffener Service genannt
- [ ] erwarteter Nutzen beschrieben
- [ ] Risiko bewertet
- [ ] Priorität festgelegt
- [ ] Aufwand grob bewertet
- [ ] Owner benannt
- [ ] Zieltermin festgelegt
- [ ] Abhängigkeiten dokumentiert
- [ ] Erfolgskriterium definiert
- [ ] Status aktuell

Checkliste Wirkung prüfen

- [ ] Maßnahme vollständig umgesetzt

- [ ] Erfolgskriterium messbar
- [ ] Kennzahl vor und nach Verbesserung verglichen
- [ ] Benutzerfeedback berücksichtigt
- [ ] Incident-Entwicklung geprüft
- [ ] Risiken neu bewertet
- [ ] Nebenwirkungen geprüft
- [ ] Dokumentation aktualisiert
- [ ] Stakeholder informiert
- [ ] weitere Maßnahme bei Bedarf erfasst
- [ ] Verbesserung abgeschlossen oder angepasst

Checkliste Maßnahmenverfolgung

- [ ] jede Maßnahme besitzt Owner
- [ ] jeder Termin ist realistisch
- [ ] Status wird regelmäßig aktualisiert
- [ ] Blockierungen sind sichtbar
- [ ] benötigte Entscheidungen sind benannt
- [ ] Risiko bei Verzögerung ist bekannt
- [ ] Wirkung wird im nächsten Review geprüft
- [ ] abgeschlossene Maßnahmen werden dokumentiert
- [ ] offene Maßnahmen werden nicht vergessen

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker liefern wichtige technische Erkenntnisse für Service Reviews und Continual Improvement.

Im Arbeitsalltag bedeutet das:

- Incident- und Monitoringdaten verständlich einordnen,
- technische Risiken früh melden,
- wiederkehrende Fehler sichtbar machen,
- Workarounds und Known Errors bewerten,
- Verbesserungen an Monitoring und Runbooks vorschlagen,
- Change-Ergebnisse dokumentieren,
- Configuration-Daten aktuell halten,
- und die Wirkung technischer Maßnahmen überprüfen.

Technische Verbesserung ist besonders wertvoll, wenn sie messbar zu stabileren Services und besserer Benutzerunterstützung führt.

Zusammenfassung

> Serviceleistung und Benutzerfeedback sammeln
> ↓
> Service Review vorbereiten
> ↓
> Ziele, Abweichungen und Risiken bewerten
> ↓
> Incidents, Problems, Changes und Lieferanten betrachten
> ↓
> Verbesserungen identifizieren
> ↓
> Nutzen, Risiko und Aufwand bewerten
> ↓
> Maßnahmen priorisieren
> ↓
> Owner, Termin und Erfolgskriterien festlegen
> ↓
> Verbesserung kontrolliert umsetzen
> ↓
> Wirkung messen
> ↓
> Ergebnis im nächsten Service Review prüfen

Merksätze

> Ein Service Review macht aus Reporting konkrete Entscheidungen.

> Grüne Kennzahlen bedeuten nicht automatisch gute Servicequalität.

> Benutzerfeedback gehört genauso zum Review wie technische Messwerte.

>

> Jede wichtige Verbesserung braucht Owner, Termin und Erfolgskriterium.

> “ Eine umgesetzte Maßnahme ist erst erfolgreich, wenn ihre Wirkung geprüft wurde.

> “ Continual Improvement besteht aus vielen kleinen und großen Verbesserungen.

> “ Reviews müssen auch kommende Risiken und Anforderungen betrachten.

Verwandte Seiten

- 8.1 Service Level Management – Ziele, Begriffe und Grundlagen
- 8.2 SLA, SLR, OLA und Underpinning Contracts
- 8.3 Kennzahlen, Messmethoden und Reporting
- 8.5 Service Level Management im Zusammenspiel mit Incident, Problem, Change und Configuration Management
- Continual Improvement
- Measurement and Reporting
- Incident Management
- Problem Management
- Change Enablement
- Supplier Management
- Knowledge Management

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Service Level Management
- PeopleCert – ITIL Practice Guide: Continual Improvement
- PeopleCert – ITIL Practice Guide: Measurement and Reporting
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Problem Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- PeopleCert – ITIL Practice Guide: Supplier Management
- ITIL Foundation – Version 5

Einordnung

Die dargestellten:

- Review-Agenda,
- Improvement-Register-Struktur,
- Priorisierungskriterien,
- Maßnahmenprotokolle,
- Checklisten,
- Erfolgskriterien,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Service-Review-Agenda,
- Review-Frequenz,
- Teilnehmerliste,
- Improvement-Register-Vorlage,
- Priorisierungsmatrix,
- Maßnahmenstruktur,
- oder Reporting-Form

für alle Organisationen vor.

Die konkrete Umsetzung muss an:

- Services,
- Kritikalität,
- Kunden,
- Benutzergruppen,
- Risiken,
- Organisationsgröße,
- Supportmodell,
- Lieferanten,
- Datenqualität,
- und verfügbare Fähigkeiten

angepasst werden.

Behandelter Framework-Stand: ITIL Version 5
Zusätzlich berücksichtigt: aktuelle ITIL-4-Practice-Guidance
Fachlicher Stand: August 2026
