4.4 Trendanalyse und proaktives Problem Management

> **Kurz erklärt**
>
> Trendanalyse und proaktives Problem Management helfen dabei, wiederkehrende Muster frühzeitig zu erkennen.
>
> Ziel ist es, Problems nicht erst nach großen Störungen zu bearbeiten, sondern mögliche Ursachen und Risiken bereits vorher sichtbar zu machen.
>
> Dadurch können Incidents reduziert, Auswirkungen begrenzt und Services langfristig stabiler betrieben werden.

Warum Trendanalyse wichtig ist

Ein einzelner Incident wirkt oft wie ein isolierter Fehler.

Erst durch die Betrachtung vieler Vorgänge wird sichtbar, ob ein Muster besteht.

Beispiele:

- VPN-Störungen nehmen jede Woche zu.
- Druckerprobleme treten immer am gleichen Standort auf.
- Anmeldefehler häufen sich nach Passwortänderungen.
- Speicherwarnungen treten regelmäßig vor Monatsende auf.
- Eine Anwendung wird nach jedem Update instabil.
- Ein bestimmter Service erzeugt überdurchschnittlich viele Tickets.

Trendanalyse hilft dabei, solche Muster zu erkennen und daraus Verbesserungen abzuleiten.

Reaktives und proaktives Problem Management unterscheiden

Reaktives Problem Management	Proaktives Problem Management
beginnt nach Incidents	sucht nach Risiken und Mustern vor größeren Incidents
reagiert auf Störungen	verhindert oder reduziert zukünftige Störungen
nutzt Incident-Daten nach einem Ereignis	nutzt Trends, Monitoring, Reviews und Erfahrungswerte
häufig nach Major Incidents	häufig durch regelmäßige Analyse
Ziel: Ursache verstehen	Ziel: Risiken frühzeitig erkennen

Beide Formen sind wichtig.

Reaktives Problem Management lernt aus bereits eingetretenen Incidents.

Proaktives Problem Management versucht, zukünftige Incidents zu vermeiden.

Was ist eine Trendanalyse?

Trendanalyse bedeutet, Daten über einen Zeitraum zu betrachten und Muster zu erkennen.

Mögliche Fragen:

- Welche Incidents treten besonders häufig auf?
- Welche Services sind überdurchschnittlich betroffen?
- Welche Standorte melden ähnliche Probleme?
- Welche Zeiträume sind auffällig?
- Welche Changes stehen zeitlich mit Incidents in Verbindung?
- Welche Kategorien wachsen stark an?
- Welche Workarounds werden häufig genutzt?
- Welche Teams erhalten besonders viele Eskalationen?
- Welche Fehler kehren trotz Lösung wieder?

Eine Trendanalyse betrachtet nicht nur einzelne Tickets, sondern Zusammenhänge.

Mögliche Datenquellen

Geeignete Quellen für Trendanalysen:

- Incident Records,
- Problem Records,
- Known Errors,
- Change Records,
- Monitoringdaten,
- Logdaten,
- Service-Desk-Kontakte,
- Benutzerfeedback,
- SLA-Berichte,
- Major-Incident-Reviews,
- Service Reviews,
- Knowledge-Base-Nutzung,
- Self-Service-Daten,
- CMDB-Informationen,
- Lieferantenberichte,
- Sicherheitsmeldungen.

Je besser die Datenqualität, desto aussagekräftiger die Analyse.

Typische Muster

Muster	Mögliche Bedeutung
viele gleiche Incidents	wiederkehrendes Problem
steigende Ticketzahlen	wachsendes Risiko oder schlechter Service
gleiche Symptome nach Changes	mögliche Change-Ursache
hohe Wiedereröffnungsquote	unvollständige Lösung
viele Eskalationen an ein Team	fehlendes Wissen im Service Desk
häufige Workarounds	dauerhafte Ursache nicht beseitigt
viele Suchanfragen ohne Treffer	fehlende Knowledge-Artikel
häufige Störung an einem Standort	lokales Infrastrukturproblem
wiederkehrende Speicherwarnungen	Kapazitäts- oder Monitoringproblem

---

Trendanalyse im Service Desk

Der Service Desk ist eine besonders wichtige Quelle für Trends.

Dort entstehen täglich Informationen über:

- häufige Benutzerprobleme,
- unklare Services,
- wiederkehrende Fehler,
- schlechte Formulare,
- unverständliche Knowledge-Artikel,
- fehlende Standardlösungen,
- unklare Zuständigkeiten,
- und wiederholte Eskalationen.

Wenn diese Informationen nur in einzelnen Tickets bleiben, gehen wichtige Verbesserungsmöglichkeiten verloren.

Zeitliche Muster erkennen

Incidents können sich zeitlich häufen.

Beispiele:

- viele Anmeldeprobleme am Montagmorgen,
- Druckerprobleme zum Schichtbeginn,

- Performance-Probleme am Monatsende,
- Speicherwarnungen nach nächtlichen Jobs,
- Backupfehler nach Wartungsfenstern,
- erhöhte Ticketzahlen nach Updates.

Solche Muster helfen bei der Ursachenfindung.

Beispiel:

Wenn ein Speicherbereich jeden Montagmorgen voll ist, kann ein Wochenendjob oder eine fehlende Bereinigung beteiligt sein.

Servicebezogene Muster erkennen

Zu prüfen ist:

- Welche Services erzeugen die meisten Incidents?
- Welche Services verursachen die längsten Ausfälle?
- Welche Services haben die meisten Wiedereröffnungen?
- Welche Services benötigen häufig Eskalation?
- Welche Services haben viele Benutzerbeschwerden?
- Welche Services besitzen viele Known Errors?

Ein Service mit wenigen, aber sehr kritischen Incidents kann wichtiger sein als ein Service mit vielen kleinen Standardanfragen.

Standortbezogene Muster erkennen

Standortbezogene Trends können Hinweise liefern auf:

- Netzwerkprobleme,
- WLAN-Abdeckung,
- lokale Druckinfrastruktur,
- defekte Hardware,
- unzureichende Schulung,
- lokale Prozesse,
- Providerprobleme,
- oder Standortbesonderheiten.

Beispiel:

Wenn nur ein Standort häufig VPN-Abbrüche meldet, muss nicht der VPN-Service selbst die Hauptursache sein.

Möglicherweise liegt das Problem bei lokaler Internetanbindung, Firewall, WLAN oder Routing.

Benutzergruppen und Rollen betrachten

Manchmal treten Incidents vor allem bei bestimmten Benutzergruppen auf.

Beispiele:

- Außendienst meldet häufig VPN- und Mobilgeräteprobleme.
- Buchhaltung meldet wiederkehrende Druck- und Exportfehler.
- Lager meldet Probleme mit Scanner- oder Etikettendruck.
- neue Mitarbeiter melden viele Zugriffsprobleme.
- Führungskräfte melden häufig Kalender- und Freigabethemen.

Solche Muster können auf fehlende Schulung, unklare Prozesse oder technische Sonderanforderungen hinweisen.

Change-bezogene Muster erkennen

Viele Problems entstehen im Zusammenhang mit Änderungen.

Zu prüfen ist:

- Häufen sich Incidents nach bestimmten Changes?
- Sind bestimmte Change-Typen besonders fehleranfällig?
- Treten Fehler nach Softwareupdates auf?
- Sind Rollbacks häufig notwendig?
- Werden Standard-Changes falsch genutzt?
- Werden Tests vor Changes ausreichend durchgeführt?
- Sind betroffene Services nach Changes korrekt überwacht?

Trenddaten können zeigen, ob Change Enablement verbessert werden muss.

Monitoringdaten proaktiv nutzen

Monitoring zeigt nicht nur akute Störungen.

Es kann auch zukünftige Risiken sichtbar machen.

Beispiele:

- Speicher wächst kontinuierlich.
- Antwortzeiten steigen über Wochen.
- Fehlerquote nimmt langsam zu.
- CPU-Auslastung ist regelmäßig nahe am Limit.
- Zertifikate laufen bald ab.
- Backupdauer überschreitet zunehmend das Zeitfenster.
- Datenbankverbindungen nähern sich der Obergrenze.

- Warteschlangen werden länger.

Solche Signale können ein Problem anzeigen, bevor Benutzer betroffen sind.

Kapazitätstrends

Kapazitätstrends helfen zu erkennen, ob Ressourcen bald nicht mehr ausreichen.

Zu betrachten sind:

- Speicherplatz,
- CPU,
- Arbeitsspeicher,
- Netzwerkbandbreite,
- Datenbankgröße,
- Anzahl gleichzeitiger Sitzungen,
- Lizenzverbrauch,
- Backupfenster,
- Cloud-Kosten,
- API-Limits.

Beispiel:

Wenn ein Dateiserver jeden Monat um 12 Prozent wächst, kann bereits vor einem Ausfall geplant werden, wie Kapazität erweitert oder Daten bereinigt werden.

Known Errors auswerten

Known Errors sollten regelmäßig überprüft werden.

Fragen:

- Wie oft tritt der Known Error noch auf?
- Wie viele Incidents sind damit verbunden?
- Funktioniert der Workaround zuverlässig?
- Gibt es inzwischen eine dauerhafte Lösung?
- Ist das Risiko noch akzeptabel?
- Sind Knowledge-Artikel aktuell?
- Muss ein Change priorisiert werden?
- Gibt es neue betroffene Versionen oder Services?

Ein Known Error darf nicht dauerhaft unbeachtet bleiben.

Workaround-Nutzung als Warnsignal

Wenn ein Workaround sehr häufig genutzt wird, ist das ein Hinweis auf ein ungelöstes Problem.

Beispiel:

Der Service Desk startet jede Woche denselben Dienst neu.

Der Workaround funktioniert kurzfristig.

Trotzdem zeigt die Häufigkeit, dass eine dauerhafte Lösung notwendig ist.

Zu prüfen ist:

- Wie oft wird der Workaround genutzt?
- Welche Benutzer sind betroffen?
- Wie viel Arbeitszeit kostet er?
- Welche Risiken entstehen?
- Welche dauerhafte Lösung wäre möglich?
- Warum wurde sie bisher nicht umgesetzt?

Knowledge-Base-Daten nutzen

Die Knowledge Base liefert Hinweise auf wiederkehrende Themen.

Mögliche Signale:

- bestimmte Artikel werden sehr häufig aufgerufen,
- Benutzer bewerten Artikel als nicht hilfreich,
- viele Suchanfragen führen zu keinem Treffer,
- Service Desk nutzt immer denselben Artikel,
- Artikel werden nach Updates schnell veraltet,
- bestimmte Themen erzeugen trotz Artikel weiterhin viele Tickets.

Daraus können Verbesserungen entstehen:

- Artikel überarbeiten,
- neue Artikel erstellen,
- Self-Service verbessern,
- Formulare anpassen,
- oder technische Ursache untersuchen.

Benutzerfeedback auswerten

Benutzerfeedback kann Trends sichtbar machen, die in technischen Daten nicht sofort erkennbar sind.

Beispiele:

- Anwendung ist langsam, aber Monitoring zeigt keinen Ausfall.
- Portal ist unverständlich, obwohl Requests formal korrekt erstellt werden.
- Benutzer nutzen E-Mail statt Self-Service, weil sie den Service nicht finden.
- Workaround funktioniert technisch, ist aber im Alltag zu umständlich.
- Statusmeldungen kommen zu spät oder sind unverständlich.

Qualitative Rückmeldungen ergänzen Kennzahlen.

Lieferanteninformationen einbeziehen

Auch Lieferanten können wichtige Hinweise liefern.

Beispiele:

- bekannte Softwarefehler,
- Firmwareprobleme,
- Sicherheitswarnungen,
- Cloud-Service-Störungen,
- Versionshinweise,
- End-of-Life-Ankündigungen,
- Supportfälle anderer Kunden,
- Roadmap-Informationen.

Ein wiederkehrender interner Incident kann mit einem bekannten Herstellerfehler zusammenhängen.

Sicherheitsmeldungen als Problem-Auslöser

Nicht jede Sicherheitsmeldung ist ein Incident.

Manche Sicherheitsmeldungen weisen auf ein Problem oder Risiko hin.

Beispiele:

- wiederholte fehlgeschlagene Anmeldeversuche,
- veraltete Softwareversionen,
- unsichere Standardkonfiguration,
- fehlende Patches,
- schwache Berechtigungsprozesse,
- wiederkehrende Phishing-Erfolge,
- nicht inventarisierte Geräte.

Proaktives Problem Management kann helfen, solche Risiken strukturiert zu bearbeiten.

Trend, Problem und Verbesserung verbinden

Ein Trend allein ist noch keine Lösung.

Ein sinnvoller Ablauf:

```
Daten sammeln
      ↓
Muster erkennen
      ↓
Trend bewerten
      ↓
Problem Record erstellen
      ↓
Ursache oder Risiko untersuchen
      ↓
Workaround oder Schutzmaßnahme bereitstellen
      ↓
dauerhafte Verbesserung planen
      ↓
Wirkung messen
```

Dadurch wird aus Beobachtung eine steuerbare Verbesserung.

Wann aus einem Trend ein Problem Record entsteht

Ein Problem Record kann sinnvoll sein, wenn:

- ein Trend wiederholt auftritt,
- viele Benutzer betroffen sind,
- ein kritischer Service betroffen ist,
- die Ursache unbekannt ist,
- ein Workaround häufig verwendet wird,
- ein Risiko steigt,
- Kosten oder Aufwand deutlich zunehmen,
- Sicherheitsrelevanz besteht,
- oder eine dauerhafte Verbesserung notwendig erscheint.

Nicht jede Auffälligkeit benötigt sofort einen Problem Record.

Aber wichtige Trends sollten nicht verloren gehen.

Priorisierung proaktiver Problems

Proaktive Problems konkurrieren mit anderen Aufgaben.

Kriterien zur Priorisierung:

- geschäftliche Auswirkung,
- Anzahl betroffener Benutzer,
- Kritikalität des Service,
- Risiko eines zukünftigen Incidents,
- Sicherheitsrelevanz,
- Häufigkeit,
- Kosten,
- Aufwand,
- Verfügbarkeit eines Workarounds,
- erwarteter Nutzen,
- Abhängigkeit von Changes oder Lieferanten.

Ein Problem ohne aktuellen Ausfall kann trotzdem hohe Priorität besitzen, wenn ein schwerer Ausfall absehbar ist.

Beispiel: Speichertrend

Beobachtung

Monitoring zeigt, dass ein Dateiserver jeden Monat stark wächst.

Trend

Bei gleicher Entwicklung ist der Speicher in sechs Wochen voll.

Proaktives Problem Management

- Ursache des Wachstums prüfen,
- Datenarten analysieren,
- Verantwortliche identifizieren,
- Bereinigungsregeln bewerten,
- Kapazitätserweiterung prüfen,
- Monitoring-Grenzen anpassen,
- Runbook erstellen.

Nutzen

Ein zukünftiger Ausfall wird verhindert.

Beispiel: Häufige Passwort- und MFA-Tickets

Beobachtung

Ein großer Anteil der Service-Desk-Kontakte betrifft Passwort und MFA.

Analyse

Viele Benutzer kennen den Self-Service-Passwort-Reset nicht.

MFA-Anleitungen sind veraltet.

Verbesserung

- Knowledge-Artikel aktualisieren,
- Portalhinweise verbessern,
- Self-Service sichtbarer machen,
- Onboarding-Unterlagen ergänzen,
- häufige Fehler im Formular erklären.

Nutzen

Weniger Standardkontakte und schnellere Hilfe für Benutzer.

Beispiel: VPN-Incidents nach Clientupdate

Beobachtung

Nach einem Clientupdate steigen VPN-Incidents deutlich an.

Analyse

Betroffen sind nur Windows-Notebooks mit einer bestimmten Version.

Problem Record

Ein Problem wird eröffnet, um Ursache, Workaround und dauerhafte Lösung zu verfolgen.

Maßnahmen

- Known Error dokumentieren,
- Workaround bereitstellen,
- Herstellerhinweis prüfen,
- neue Version testen,
- Rollout über Change Enablement planen.

Beispiel: Eskalationen an Netzwerkteam steigen

Beobachtung

Immer mehr Tickets werden vom Service Desk an das Netzwerkteam eskaliert.

Analyse

Viele Eskalationen betreffen einfache DNS- und VPN-Prüfungen.

Verbesserung

- Runbook für Erstdiagnose erstellen,
- Service Desk schulen,
- Knowledge-Artikel ergänzen,
- Diagnosewerkzeug bereitstellen,
- Eskalationskriterien präzisieren.

Nutzen

Weniger unnötige Eskalationen und schnellere Bearbeitung.

Typische Fehler

Fehler 1

Trends werden nicht ausgewertet, obwohl Daten vorhanden sind.

Fehler 2

Einzelne Incidents werden gelöst, aber Muster bleiben unbeachtet.

Fehler 3

Nur technische Kennzahlen werden betrachtet.

Fehler 4

Benutzerfeedback wird ignoriert.

Fehler 5

Monitoring wird nur für Alarmierung genutzt, nicht für proaktive Analyse.

Fehler 6

Known Errors bleiben offen, ohne Häufigkeit und Risiko neu zu bewerten.

Fehler 7

Trends werden erkannt, aber keine Maßnahmen abgeleitet.

Fehler 8

Proaktive Problems werden immer niedriger priorisiert als akute Tickets.

Fehler 9

Datenqualität ist schlecht, wird aber nicht verbessert.

Fehler 10

Verbesserungsmaßnahmen werden nicht auf Wirksamkeit geprüft.

Fehler 11

Lieferantenhinweise und bekannte Herstellerfehler werden nicht berücksichtigt.

Fehler 12

Trendanalysen werden nur einmalig durchgeführt statt regelmäßig.

Checkliste Trendanalyse

- [ ] relevante Datenquellen festgelegt
- [ ] Zeitraum definiert
- [ ] Incidents nach Service ausgewertet
- [ ] Incidents nach Kategorie ausgewertet
- [ ] Standorte und Benutzergruppen betrachtet
- [ ] Wiederholungen erkannt
- [ ] Changes zeitlich verglichen
- [ ] Monitoringdaten berücksichtigt
- [ ] Known Errors ausgewertet
- [ ] Workaround-Nutzung geprüft
- [ ] Benutzerfeedback berücksichtigt
- [ ] auffällige Trends dokumentiert

Checkliste proaktives Problem Management

- [ ] Trend oder Risiko beschrieben
- [ ] betroffener Service bekannt
- [ ] mögliche Auswirkungen bewertet
- [ ] Priorität festgelegt
- [ ] Problem Record bei Bedarf erstellt
- [ ] Verantwortlicher benannt
- [ ] Daten und Fakten gesammelt
- [ ] Hypothesen formuliert
- [ ] Workaround oder Schutzmaßnahme geprüft
- [ ] dauerhafte Lösung bewertet
- [ ] Change-Bedarf geprüft
- [ ] Verbesserung verfolgt
- [ ] Wirksamkeit geprüft

Checkliste Datenqualität

- [ ] Tickets enthalten betroffenen Service
- [ ] Kategorien werden einheitlich verwendet
- [ ] Prioritäten sind begründet
- [ ] Workarounds sind dokumentiert
- [ ] Changes sind nachvollziehbar verknüpft
- [ ] Monitoringdaten sind verfügbar
- [ ] Zeitstempel sind korrekt
- [ ] CMDB-Informationen sind ausreichend aktuell
- [ ] Known Errors sind auffindbar
- [ ] Abschlussnotizen sind verständlich

Bedeutung für Fachinformatiker für Systemintegration

Fachinformatiker erkennen Trends häufig direkt im technischen Alltag.

Beispiele:

- gleiche Fehlermeldung taucht immer wieder auf,
- derselbe Server erzeugt regelmäßig Warnungen,

- bestimmte Geräteklasse verursacht viele Tickets,
- ein Workaround wird ständig wiederholt,
- ein Monitoringwert verschlechtert sich langsam,
- oder ein Standort meldet auffällig viele Netzwerkprobleme.

Wichtig ist, solche Beobachtungen nicht nur mündlich weiterzugeben, sondern nachvollziehbar zu dokumentieren.

So können daraus Problems, Known Errors, Changes oder Verbesserungen entstehen.

Zusammenfassung

> Incidents, Monitoring und Feedback auswerten
> ↓
> Muster und Trends erkennen
> ↓
> Risiko und Auswirkung bewerten
> ↓
> Problem Record bei Bedarf erstellen
> ↓
> Ursache oder mögliche Ursache untersuchen
> ↓
> Workaround oder Schutzmaßnahme bereitstellen
> ↓
> dauerhafte Verbesserung planen
> ↓
> Change, Knowledge oder Continual Improvement einbinden
> ↓
> Wirkung messen und Datenqualität verbessern

Merksätze

> Trends zeigen, wo wiederkehrende Ursachen verborgen sein können.

> Proaktives Problem Management beginnt, bevor der große Ausfall eintritt.

>

> Monitoring ist nicht nur Alarmierung, sondern auch Frühwarnsystem.

> Häufig genutzte Workarounds sind ein Hinweis auf ungelöste Problems.

> Gute Datenqualität ist die Grundlage guter Trendanalyse.

> Ein erkannter Trend ist erst wertvoll, wenn daraus eine konkrete Verbesserung entsteht.

Verwandte Seiten

- 4.1 Problem Management – Ziele, Begriffe und Abgrenzung
- 4.2 Ursachenanalyse
- 4.3 Workarounds und Known Errors
- 4.5 Problem Management im Zusammenspiel mit Incident, Change und Knowledge Management
- 3.9 Kennzahlen, Qualität und Continual Improvement im Service Desk
- Service Level Management
- Knowledge Management
- Change Enablement
- Continual Improvement
- Measurement and Reporting

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

Einordnung

Die dargestellten:

- Trendanalyse-Beispiele,
- Datenquellen,
- Prüffragen,
- Checklisten,
- Priorisierungskriterien,
- und Praxisbeispiele

sind herstellerneutrale Praxisempfehlungen.

ITIL schreibt keine universelle:

- Trendanalyse-Methode,
- Datenquellenliste,
- KPI-Struktur,
- Problem-Eröffnungsschwelle,
- Analysefrequenz,
- oder Toolauswahl

für alle Organisationen vor.

Die konkrete Umsetzung muss an:

- Services,
- Risiken,
- Datenqualität,
- Monitoring,
- Supportmodell,
- Service Levels,
- Organisation,
- Lieferanten,
- und verfügbare Fähigkeiten

angepasst werden.

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