10.5 Zusammenspiel mit Incident, Problem, Change und Service Configuration Management

> **Kurz erklärt**
>
> Monitoring and Event Management arbeitet eng mit anderen ITIL Practices zusammen.
>
> Monitoring erkennt Ereignisse, bewertet deren Bedeutung und stellt Informationen für andere Prozesse bereit.
>
> Erst durch das Zusammenspiel mit Incident Management, Problem Management, Change Enablement und Service Configuration Management entsteht ein vollständiger und effizienter IT-Servicebetrieb.

Warum das Zusammenspiel wichtig ist

Monitoring allein behebt keine Fehler.

Es beantwortet zunächst nur die Frage:

> **"Was ist passiert?"**

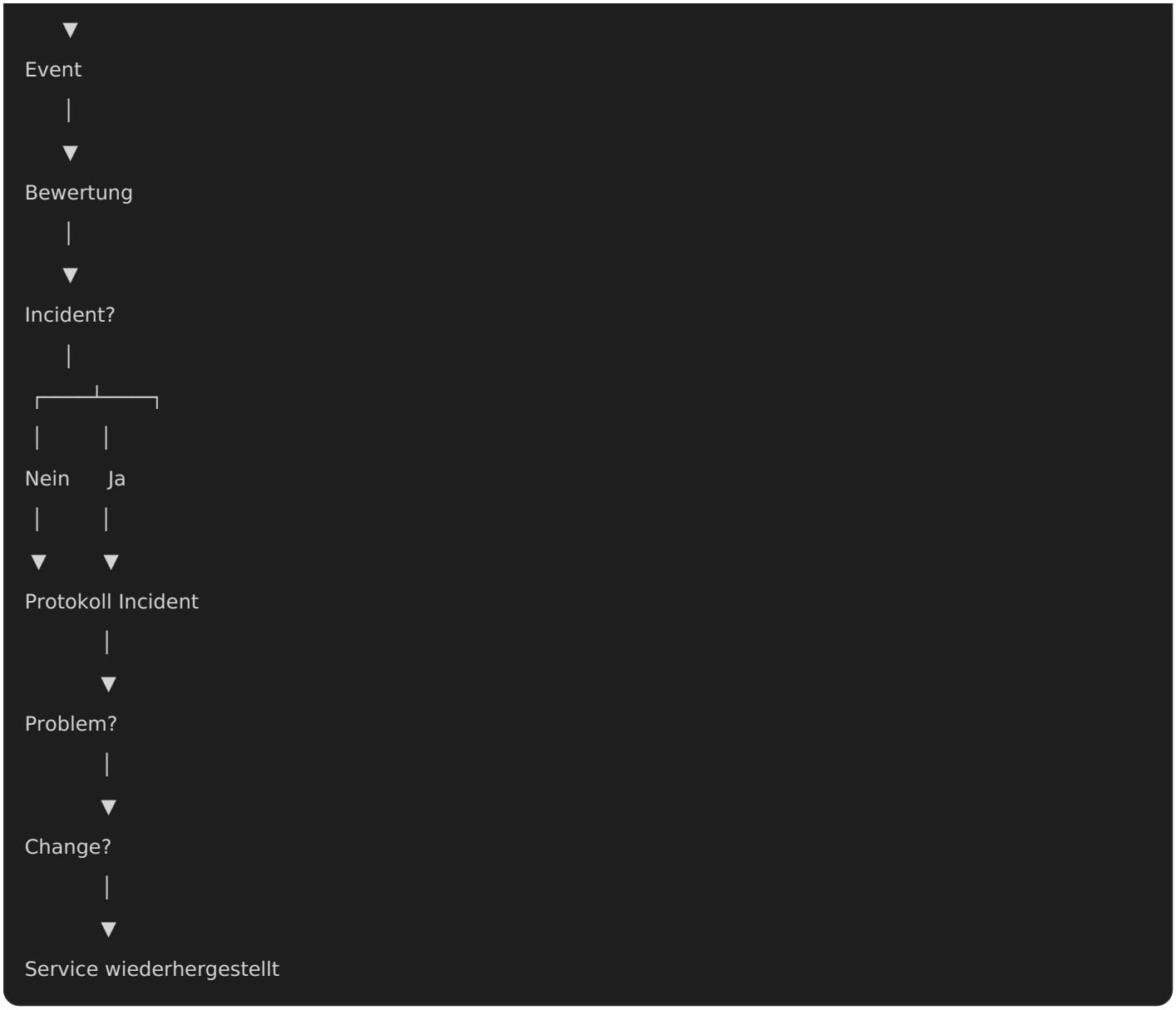
Andere ITIL Practices beantworten anschließend beispielsweise:

- Muss eine Störung behoben werden?
- Ist die Ursache bekannt?
- Muss eine Änderung durchgeführt werden?
- Welche Systeme sind betroffen?
- Welche Services sind beeinträchtigt?

Erst gemeinsam entsteht ein vollständiger Ablauf.

Zusammenspiel der Practices

```
Monitoring
    |
```



Nicht jedes Event durchläuft alle Schritte.

Monitoring und Incident Management

Diese beiden Practices arbeiten besonders eng zusammen.

Monitoring erkennt Ereignisse.

Incident Management stellt den normalen Servicebetrieb wieder her.

Beispiel:

Monitoring erkennt:

> “ Webserver antwortet nicht.

Anschließend:

- Incident erstellen,
- Service Desk informieren,
- Administrator alarmieren,
- Service wiederherstellen.

Monitoring erkennt den Fehler.

Incident Management behebt ihn.

Monitoring erkennt häufig Incidents vor dem Benutzer

Ein großer Vorteil moderner Monitoring-Systeme:

Probleme werden oft erkannt,

bevor Benutzer sie melden.

Beispiele:

- Speicher fast voll
- Backup fehlgeschlagen
- Zertifikat läuft bald ab
- RAID degradiert
- Datenbank antwortet langsam

Viele Incidents können dadurch vollständig vermieden werden.

Monitoring und Problem Management

Ein einzelner Alarm ist häufig kein Problem.

Wiederholen sich jedoch dieselben Events regelmäßig,

kann daraus ein Problem Record entstehen.

Beispiel:

Jeden Dienstag:

- Datenbank langsam.

Incident wird jedes Mal behoben.

Monitoring zeigt jedoch,

dass dieselbe Ursache regelmäßig auftritt.

Jetzt beginnt Problem Management mit der Ursachenanalyse.

Trendanalysen unterstützen Problem Management

Monitoring speichert historische Daten.

Dadurch lassen sich erkennen:

- wiederkehrende Fehler,
- steigende Auslastung,
- häufige Neustarts,
- zunehmende Antwortzeiten,
- regelmäßig fehlschlagende Backups.

Diese Informationen helfen bei der Suche nach der eigentlichen Ursache.

Monitoring und Change Enablement

Viele Changes beeinflussen Monitoring.

Beispiele:

- neuer Server,
- zusätzliche VM,
- neue Firewall,
- neue Anwendung,
- Cloud-Service,
- Netzwerksegment.

Nach einem Change muss häufig auch das Monitoring angepasst werden.

Monitoring vor einem Change

Vor Änderungen liefern Monitoring-Daten wichtige Informationen.

Beispiele:

- aktuelle Auslastung,
- Antwortzeiten,
- Verfügbarkeit,
- Fehlerquote,
- Baseline.

Dadurch kann später geprüft werden,

ob der Change erfolgreich war.

Monitoring nach einem Change

Nach der Umsetzung wird kontrolliert:

- Service erreichbar?
- Fehler aufgetreten?
- Antwortzeiten verändert?
- CPU erhöht?
- neue Alarme?

Monitoring bestätigt,

ob die Änderung erfolgreich war.

Geplante Wartungen

Während eines genehmigten Changes sollten Monitoring-Systeme dies berücksichtigen.

Beispiel:

Server wird bewusst neu gestartet.

Ohne Wartungsfenster:

- Alarm,
- Incident,
- Eskalation.

Mit Wartungsfenster:

- Event wird dokumentiert,
- keine unnötigen Alarme entstehen.

Monitoring und Service Configuration Management

Monitoring überwacht Configuration Items.

Configuration Management beschreibt diese.

Beispiele:

- Server,
- Switch,
- Router,
- Datenbank,
- Container,
- VM,
- Storage,
- Firewall.

Monitoring liefert aktuelle Zustandsinformationen.

Configuration Management liefert Struktur und Beziehungen.

Warum Configuration Items wichtig sind

Ohne Configuration Management weiß Monitoring häufig nicht,

welche Bedeutung ein System besitzt.

Beispiel:

Server 01 fällt aus.

Monitoring erkennt den Ausfall.

Erst Configuration Management zeigt:

Dieser Server gehört zum:

- ERP-System,
- Produktionssteuerung,
- Rechnungswesen.

Dadurch lässt sich die Auswirkung deutlich besser bewerten.

Service Maps

Service Maps verbinden Monitoring und Configuration Management.

Beispiel:

```
Online-Shop
    |
    ├── Webserver
```

```
├── Load Balancer
├── Datenbank
├── Storage
├── DNS
└── Payment API
```

Fällt eine Komponente aus,

werden die betroffenen Services sofort sichtbar.

Abhängigkeiten erkennen

Configuration Management zeigt,

welche Systeme voneinander abhängig sind.

Dadurch kann Monitoring:

- Alarme korrelieren,
- Prioritäten anpassen,
- Root Causes schneller erkennen.

Monitoring und Availability Management

Monitoring liefert die Grundlage zur Messung der Verfügbarkeit.

Beispiele:

- Server erreichbar?
- Anwendung verfügbar?
- API antwortet?
- Website erreichbar?

Availability Management nutzt diese Daten,

um Verfügbarkeitsziele zu bewerten.

Monitoring und Capacity Management

Monitoring misst kontinuierlich:

- CPU,
- RAM,
- Speicher,

- Netzwerk,
- Datenbankgröße.

Capacity Management nutzt diese Informationen,

um zukünftige Ressourcen zu planen.

Monitoring und Information Security Management

Monitoring erkennt auch sicherheitsrelevante Ereignisse.

Beispiele:

- ungewöhnliche Logins,
- fehlgeschlagene Anmeldungen,
- Malware,
- Firewall-Ereignisse,
- verdächtiger Netzwerkverkehr,
- Privilegienänderungen.

Diese Events können an Security-Prozesse weitergeleitet werden.

Monitoring und Continual Improvement

Monitoring liefert kontinuierlich Kennzahlen.

Dadurch lassen sich Verbesserungen erkennen.

Beispiele:

- MTTR sinkt,
- weniger Incidents,
- höhere Verfügbarkeit,
- kürzere Antwortzeiten,
- bessere Automatisierung.

Continual Improvement nutzt diese Daten,

um Prozesse weiterzuentwickeln.

Monitoring und Service Level Management

Viele SLA-Kennzahlen basieren direkt auf Monitoring-Daten.

Beispiele:

- Verfügbarkeit,
- Antwortzeit,
- Ausfallzeit,
- Servicequalität.

Ohne zuverlässiges Monitoring können SLA-Ziele kaum objektiv gemessen werden.

Praxisbeispiel

Das Monitoring erkennt:

Die Antwortzeit einer Datenbank steigt seit mehreren Wochen kontinuierlich an.

Zunächst entstehen einzelne Warning Events.

Später treten wiederholt Incidents auf.

Problem Management untersucht die Ursache und stellt fest,

dass der verfügbare Speicher nicht mehr ausreicht.

Ein genehmigter Change erweitert den Speicher.

Nach der Umsetzung zeigt das Monitoring wieder normale Antwortzeiten.

Die gewonnenen Erkenntnisse fließen anschließend in Continual Improvement ein.

Typische Fehler

Fehler 1

Monitoring arbeitet unabhängig von Incident Management.

Fehler 2

Configuration Items sind nicht aktuell.

Fehler 3

Changes berücksichtigen das Monitoring nicht.

Fehler 4

Monitoring-Daten werden nicht ausgewertet.

Fehler 5

Wiederkehrende Events führen nicht zu Problem Records.

Fehler 6

Service Maps fehlen.

Fehler 7

Abhängigkeiten werden nicht berücksichtigt.

Fehler 8

Monitoring misst keine SLA-Kennzahlen.

Fehler 9

Kapazitätsdaten werden nicht ausgewertet.

Fehler 10

Verbesserungen basieren nicht auf Messdaten.

Checkliste Zusammenspiel

- [ ] Monitoring aktiv
- [ ] Event korrekt bewertet
- [ ] Incident bei Bedarf erstellt
- [ ] Problem Record bei wiederkehrenden Fehlern geprüft
- [ ] Change berücksichtigt
- [ ] Configuration Items aktuell
- [ ] Service Maps gepflegt
- [ ] SLA-Kennzahlen verfügbar
- [ ] Sicherheitsereignisse erkannt
- [ ] Verbesserungspotenziale dokumentiert

Checkliste Monitoring-Prozess

- [ ] Events gesammelt
- [ ] Filter eingerichtet
- [ ] Alarmregeln geprüft
- [ ] Eskalationen definiert
- [ ] Dashboards aktuell
- [ ] KPIs überwacht
- [ ] Trends ausgewertet
- [ ] Reports erstellt
- [ ] Verantwortlichkeiten bekannt
- [ ] regelmäßige Optimierung geplant

Bedeutung für Fachinformatiker für Systemintegration

Monitoring gehört zu den wichtigsten Werkzeugen im Arbeitsalltag eines Fachinformatikers.

Typische Aufgaben:

- Monitoring konfigurieren,
- Alarme analysieren,
- Incidents früh erkennen,
- Monitoring nach Changes anpassen,
- Configuration-Daten pflegen,
- Service Maps aktualisieren,
- Trends auswerten,
- Automatisierungen verbessern.

Ein gut integriertes Monitoring unterstützt nahezu alle Bereiche des IT-Service-Managements.

Zusammenfassung

> Monitoring erkennt Events
>
> ↓
>
> Event bewerten
>
> ↓
>
> Incident auslösen (falls erforderlich)

> ↓
>
> Problem analysieren
>
> ↓
>
> Change umsetzen
>
> ↓
>
> Monitoring überprüft Ergebnis
>
> ↓
>
> Erkenntnisse für Continual Improvement nutzen

Merksätze

> Monitoring erkennt Ereignisse – andere Practices reagieren darauf.

> Monitoring liefert die Grundlage für Incident, Problem und Change Management.

> Configuration Management erklärt die Beziehungen zwischen den überwachten Systemen.

> Gute Service Maps beschleunigen die Ursachenanalyse.

> Monitoring-Daten bilden die Grundlage für kontinuierliche Verbesserungen.

Verwandte Seiten

- 10.1 Ziele und Grundlagen des Monitoring and Event Management
- 10.2 Event-Typen, Filter und Priorisierung

- 10.3 Alarme, Eskalationen und Automatisierung
- 10.4 Monitoring-Werkzeuge, Dashboards und Kennzahlen
- Incident Management
- Problem Management
- Change Enablement
- Service Configuration Management
- Service Level Management
- Availability Management
- Capacity and Performance Management
- Information Security Management
- Continual Improvement

Quellen und Versionsstand

Offizielle Grundlagen

- PeopleCert – ITIL Practice Guide: Monitoring and Event Management
- PeopleCert – ITIL Practice Guide: Incident Management
- PeopleCert – ITIL Practice Guide: Problem Management
- PeopleCert – ITIL Practice Guide: Change Enablement
- PeopleCert – ITIL Practice Guide: Service Configuration Management
- PeopleCert – ITIL Practice Guide: Availability Management
- PeopleCert – ITIL Practice Guide: Capacity and Performance Management
- PeopleCert – ITIL Practice Guide: Service Level Management
- PeopleCert – ITIL Practice Guide: Continual Improvement
- ITIL Foundation – Version 5

Einordnung

Die dargestellten Zusammenhänge orientieren sich an den offiziellen ITIL-Practices. Monitoring and Event Management stellt Informationen für zahlreiche weitere Practices bereit, übernimmt jedoch selbst weder die Fehlerbehebung noch die Ursachenanalyse oder die Planung von Änderungen. Die konkrete technische Umsetzung hängt von den eingesetzten Monitoring-, ITSM- und CMDB-Systemen der jeweiligen Organisation ab.

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