

O201 OPERATE · PRACTITIONER

AI in IT Operations

From telemetry and service level objectives to automated diagnosis and governed remediation.

FORMAT

Knowledge pack · 13 slides

FOR

Operations and SRE leaders

LEVEL

Practitioner



THE PREMISE

Automation depends on the quality of its signals and runbooks

Noisy alerts, missing telemetry and undocumented dependencies **mislead any model** that relies on them. Start with clean signals, clear objectives and runbooks a machine can execute.

Signal

Telemetry that describes what users experience.

Objective

Targets agreed with the business.

Runbook

Steps precise enough for a machine to follow.

FOUNDATION

The operations data foundation

Telemetry

OpenTelemetry traces, metrics and logs that share service.name, so they can be joined.

Topology

Services, infrastructure and owners, with dependencies derived from traces.

Change

Deployments, configuration changes and feature flags, with timestamps.

Knowledge

Runbooks, past incidents and known errors, in machine-readable form.

Objectives

SLOs and error budgets for critical user journeys.

JOIN CHANGE DATA FIRST

Google's SRE book reports that "roughly 70% of outages are due to changes in a live system." Correlate symptoms with recent deployments and configuration changes.

OBJECTIVES

Define reliability the way users experience it

Definitions adapted from Google's Site Reliability Engineering books.

Service level indicator

A carefully defined quantitative measure of some aspect of the service provided, such as the share of requests that succeed.

Service level objective

A target for an indicator over a window, such as 99.9% of requests succeeding over 28 days.

Error budget

100% minus the objective. In Google's example policy, a spent budget halts all changes and releases except P0 and security fixes.

AI IN OPERATIONS

Where AI helps, and the risks to watch

TASK	WHAT WORKS	WATCH FOR
Detection	Learned seasonal baselines for signals that have no SLO	Paging on anomalies with no user impact
Correlation	Grouping alerts by topology, time and change	Stale topology that groups the wrong alerts
Diagnosis	Querying logs and traces; ranking recent changes as suspects	Conclusions without links to evidence
Remediation	Running approved runbook steps	Wide blast radius; untested rollback
Learning	Drafting summaries and reviews; clustering recurring causes	Drafts from thin records; summaries in place of analysis

WHAT THE EVIDENCE SAYS

Published results are strong on summaries and modest on root cause

Two production systems and one 2026 benchmark, each measured differently and not directly comparable.

51%

less time to draft each incident summary when an LLM wrote the first draft, across 300 summaries in Google's security incident response. A person reviews every draft.

Google Security Blog, April 2024

42%

of investigations in Meta's web monorepo had the root cause among the top five code changes its system suggested at the start, in backtesting.

Meta Engineering, June 2024

25.3%

best root-cause accuracy across five frontier agents on medium-difficulty on-call tasks, and 10.0% on hard ones, on a six-day, 50 GB testbed.

ORCA-bench, arXiv preprint, 2026

Both production teams withhold weak output: Google does not call the model when the input is under 200 tokens, and Meta suppresses low-confidence suggestions.

AUTONOMY

Grant remediation autonomy in levels

Promote each action, not each agent, on its track record at the level below.

Inform

The agent reports a diagnosis with its evidence.

1

Recommend

The agent proposes specific runbook steps.

2

Execute on approval

A person approves each action before it runs.

3

Execute and notify

Reversible, pre-approved standard changes run; the owner is notified.

4

Autonomous

Actions run within policy and error budget; people sample outcomes.

5

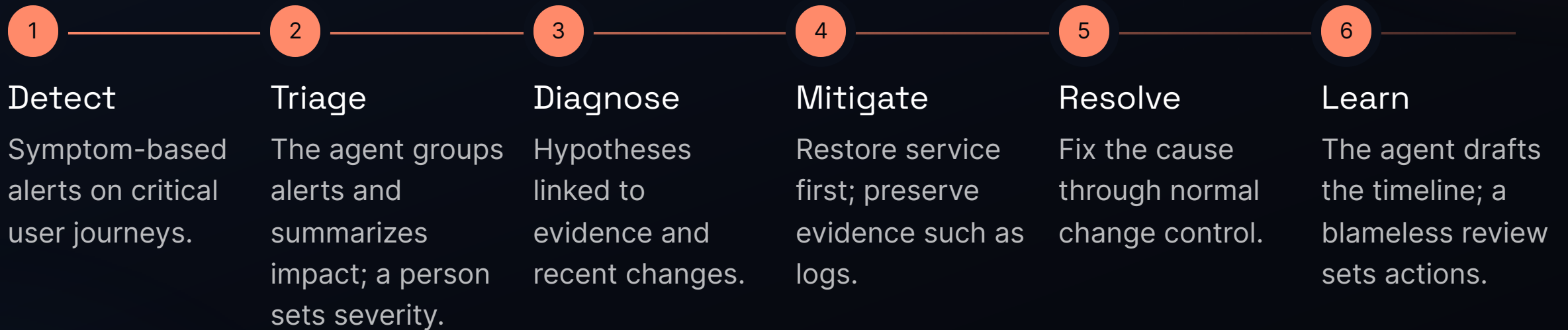
GUARDRAILS

Guardrails for automated remediation

- ✓ **Allow-list**
Only named actions on named services.
- ✓ **Blast-radius limits**
One instance or zone at a time.
- ✓ **Rate limits and circuit breakers**
Caps per service per hour; a halt after repeated failures.
- ✓ **Kill switch**
One control that stops all automation.
- ✓ **Scoped credentials**
Separate, minimal permissions for each action.
- ✓ **Automatic rollback**
Revert when indicators degrade after an action.
- ✓ **Change records**
Created automatically for every action.
- ✓ **Audit trail**
Who or what acted, when, why and with what result.

INCIDENTS

The incident lifecycle with agents



MEASUREMENT

Measure what operations delivers

Objective attainment

Objectives met for each service, and the rate at which error budget burns.

Time to restore

Review long incidents individually; durations are skewed, so averages mislead.

Change fail rate

Share of deployments that need immediate intervention, such as a rollback or hotfix.

Alert precision

Share of alerts that required action.

Toil

Share of time on manual, repetitive work; Google SRE aims to keep it below 50%.

Diagnosis accuracy

Share of automated diagnoses that the incident review confirms.

PITFALLS

Common failures in AIOps programs

✗ Alerting on causes, not symptoms

✓ Page on SLO burn, such as 2% of budget in an hour

✗ Automating unreliable runbooks

✓ Make runbooks reliable by hand first

✗ Summaries without evidence

✓ Link every conclusion to its source

✗ Steering by mean time to restore

✓ Rank incidents by error budget consumed

✗ Administrator rights for agents

✓ Scope credentials to each action

✗ Skipping the incident review

✓ Review any incident that uses over 20% of the budget

GETTING STARTED

A 90-day plan

DAYS 1–30

Signal

- ◆ Instrument the most critical services with OpenTelemetry
- ◆ Define SLOs for five critical user journeys
- ◆ Connect change data; record a baseline

DAYS 31–60

Knowledge

- ◆ Codify and test the ten most-used runbooks
- ◆ Run agent diagnosis in shadow mode
- ◆ Measure diagnosis accuracy

DAYS 61–90

Autonomy

- ◆ Add guardrails and a kill switch
- ◆ Move three reversible actions to Execute on approval
- ◆ Review results against the baseline

SUMMARY

Key takeaways

- 01 Clean telemetry, current topology and change data come before any model.
- 02 Define reliability with indicators, objectives and error budgets that reflect users.
- 03 Apply AI where it improves on static rules, and require evidence for every diagnosis.
- 04 Grant autonomy action by action, with guardrails and automatic rollback.
- 05 Measure incidents by error budget consumed, and review each significant one.

NEXT COURSE · O301

Building and Running AI Agents in Production

Operate · Advanced