

Site Reliability Engineering (SRE) Training Course

18 – 22 October 2027
Amsterdam

AGILE LEADERS
Training Center

AGILE LEADERS
Training Center

Site Reliability Engineering (SRE) Training Course

Ref: 1831_237107 **Date:** 18 – 22 October 2027 **Location:** Amsterdam **Fees:** 5700 Euro

Overview

Site Reliability Engineering SRE Training Course is a one-week intermediate course for reliability, platform, DevOps, cloud operations, systems engineering, and technical leadership functions who leave with a Production Service Reliability Plan. The course connects user-centered service levels, error budgets, observability, toil reduction, incident response, capacity engineering, automation, and reliability governance. Participants convert production evidence into reviewable operating decisions without depending on a specific cloud or monitoring product. Agile Leaders Training Center delivers this course on site reliability engineering for production services.

Who Should Attend

- Functions responsible for production-service reliability and availability decisions
- Teams responsible for platform operations, deployment safeguards, and service health
- Groups responsible for monitoring, alerting, on-call response, and incident coordination
- Personnel responsible for capacity forecasts, performance risks, and recovery readiness
- Technical leaders responsible for reliability priorities, policies, and improvement work

The course assumes participants already interpret production metrics, service dependencies, deployment records, or incident evidence and leaves out introductory coding, cloud-product administration, MLOps, and tool-specific certification preparation.

Departments and Industries

The course supports departments and industries that operate customer-facing or business-critical digital services.

- Site reliability, platform engineering, and DevOps
- Cloud operations, infrastructure, and network services
- Software engineering, application support, and service management
- Financial services, telecommunications, and digital commerce
- Healthcare technology, transport systems, and public digital services

Learning Objectives

By the end of this course, participants will be able to:

- Build service-level indicators and objectives around user journeys.
- Apply error budgets to release and reliability decisions.
- Analyze observability evidence and design actionable alerts.
- Prioritize toil reduction and reliability automation work.
- Diagnose incidents and build learning-focused response controls.
- Evaluate capacity, readiness, and reliability improvement priorities.

Course Agenda

Day 1: Define Reliability

- Service Boundary and User Journey Map
- Service-Level Indicator Selection Worksheet
- Service-Level Objective Design Record
- Error Budget Calculation and Policy Canvas
- Reliability Risk and Stakeholder Trade-Off Register

Day 2: Observe Service Health

- Metrics, Logs, Traces, and Events Evidence Map
- Symptom-Based Monitoring Design Method
- Error Budget Burn-Rate Alert Matrix
- Alert Quality and Actionability Review Checklist
- Service Dependency and Failure-Domain Diagram

Day 3: Control Operational Work

- Toil Identification and Measurement Register
- Automation Value and Risk Prioritization Matrix
- On-Call Load and Escalation Path Review
- Runbook Decision Flow and Verification Checklist
- Safe Change, Rollback, and Release Guardrail Map

Day 4: Respond and Prepare

- Incident Command Role and Communication Model
- Severity, Escalation, and Stakeholder Update Matrix
- Blameless Post-Incident Review Method
- Demand Forecast and Capacity Margin Model
- Production Readiness Review Evidence Pack

Day 5: Practice Reliability Engineering

- Suggested Exercise: Define SLIs, SLOs, and an Error Budget
- Suggested Exercise: Challenge Alerts with Burn-Rate Evidence
- Suggested Exercise: Prioritize a Toil Automation Backlog
- Suggested Exercise: Run an Incident and Post-Incident Review
- Capstone Exercise: Build the Production Service Reliability Plan

Practical Exercises

The course uses suggested activities to turn service evidence into traceable reliability decisions.

- Suggested activity: map a payment or booking journey and define measurable good and bad events.
- Suggested activity: test alert thresholds against user impact, error-budget consumption, and response ownership.
- Suggested activity: classify repetitive operational work and rank automation candidates by value and risk.
- Suggested activity: assess capacity, incident learning, readiness gaps, and improvement priorities for a production service.

FAQs

Who suits Site Reliability Engineering training, and what does it assume?

Site Reliability Engineering training suits production reliability, platform, DevOps, cloud operations, systems engineering, and technical leadership functions that already use service metrics, deployment evidence, dependency records, or incident information.

How does Site Reliability Engineering differ from general DevOps training?

Site Reliability Engineering focuses on measurable reliability targets, error budgets, operational load, incident learning, capacity, and production readiness, while general DevOps training usually covers a broader delivery culture, automation pipeline, and collaboration scope.

What are service-level indicators and objectives in SRE?

Service-level indicators measure aspects of user-experienced service performance, while service-level objectives set the target performance expected over a defined evaluation period.

How do error budgets guide SRE decisions?

Error budgets quantify the tolerated gap between perfect service and the chosen objective, giving product and operations teams a shared basis for balancing release risk with reliability work.

AGILE LEADERS
Training Center

How does SRE reduce operational toil?

SRE reduces operational toil by identifying repetitive, manual, automatable work, measuring its burden, removing avoidable demand, and prioritizing safe automation that creates lasting engineering value.

Conclusion

Participants take back a Production Service Reliability Plan. The plan links service boundaries, SLIs, SLOs, error budgets, observability, alerts, toil, incident controls, capacity evidence, readiness findings, and prioritized improvement work. It changes fragmented operational evidence into a shared basis for reliability decisions, accountable engineering action, and sustained review.