
API Testing Strategy Training Course: Risk-Based Coverage, Automation and Release Gates

10 – 14 May 2027
Amsterdam





API Testing Strategy Training Course: Risk-Based Coverage, Automation and Release Gates

Ref: 300_20680 **Date:** 10 – 14 May 2027 **Location:** Amsterdam **Fees:** 4800 GBP

Programme Summary

API testing strategy is the subject of this five-day training course for QA managers, test leads and engineering and delivery managers, ending in an API testing strategy and release gate plan for one product area. Interfaces change often and are consumed by many teams, yet test effort is usually set team by team. The approach moves from risk and test levels, through coverage and defect diagnosis, to automation and pipeline design, then to release gates and a worked plan. Mars Training Centre UK delivers this programme in API testing strategy.

The Business Case

Interface defects reach customers and partners when coverage is uneven, test data is unreliable or release decisions rest on opinion. Managers who tie test effort to risk, decide what is automated and agree what a release must pass can cut escaped defects and avoid wasted effort on checks that add little evidence.

- Escaped Defects: Fall as risk-based coverage targets the interfaces and cases that fail most often.
- Release Cycle Time: Shortens as automated checks and agreed gate criteria replace repeated manual sign-off.
- Test Maintenance Effort: Reduces as unstable checks and duplicated cases are identified and removed.

Principles and Evidence Base

- ISO/IEC/IEEE 29119: Its software testing series, including the test processes of part 2, structures the test strategy and planning work in Sessions 1 and 4.
- Test Pyramid: This model of how many checks belong at each level guides the automation choices in Session 3.
- Pareto Analysis: The observation that a few components hold most defects directs the diagnosis work in Session 2.

Intended Participants

- QA managers who set test approach and coverage rules across teams.
- Test leads who plan, automate and report interface checks.
- Engineering managers who own the release quality of their services.
- Delivery managers who decide whether a release may go ahead.

The programme serves banking, telecommunications, retail and public-sector digital teams.



Learning Outcomes

1. Participants will be able to prioritise interfaces and test cases by risk using a coverage matrix.
2. Participants will be able to diagnose coverage gaps, defect clusters and unreliable test data in an existing test suite.
3. Participants will be able to evaluate which checks to automate and where they sit in a delivery pipeline.
4. Participants will be able to construct a contract test plan for interfaces shared between teams.
5. Participants will be able to justify release gate criteria and quality indicators to a delivery sponsor.
6. Participants will be able to assign test ownership and rules for managing test debt.

Programme Structure

Day 1: API Testing Strategy: Scope, Risk and Test Levels

What must be tested, at which level, and why?

- Interface Inventory and Risk Register
- ISO/IEC/IEEE 29119 Test Process Overview
- Test Pyramid Mapping for Interfaces
- Test Strategy Document Outline
- Case Review of Two Sector Release Failures

Day 2: Coverage and Defect Diagnosis

Where are the gaps in the current tests, and where do defects cluster?

- Risk-Based Coverage Matrix
- Defect Clustering by Pareto Analysis
- Functional and Negative Case Review
- Test Data and Environment Readiness Audit
- Contract Drift Check

Day 3: Automation and Pipeline Design

Which checks should be automated, and where should they run?

- Automation Candidate Scoring Grid
- Contract Test Plan
- Pipeline Stage Placement of Interface Checks
- Performance and Security Check Selection
- Test Report Template



Day 4: Release Gates, Measures and Ownership

What must a release pass, and who maintains the tests?

- Release Gate Criteria Sheet
- Escape Rate and Flaky Test Indicators
- Test Ownership and Review Cadence
- Test Debt Backlog Rules
- Tool Selection Checklist

Day 5: Applying API Testing Strategy to One Product Area

Is the plan ready for the delivery sponsor to approve?

- Applying the Session 1 Risk Register to the Chosen Area
- Testing the Session 2 Coverage Matrix Against a Sample Suite
- Checking Session 3 Pipeline Placement Against Release Needs
- Applying the Session 4 Gate Criteria to Open Issues
- Completing the API Testing Strategy and Release Gate Plan

Methodology

The programme is built around structured exercises on an anonymised product area and a sample test suite. Participants rank risks, build coverage matrices, score automation candidates and draft gate criteria in small teams, then defend their plan before a simulated release board. No programming is required.

- Suggested activity: Building a risk-based coverage matrix for eight sample interfaces.
- Suggested activity: Scoring twelve existing checks for automation and removal.
- Suggested activity: Drafting release gate criteria for a payment-related interface change.

Capstone Output

Each participant completes an API testing strategy and release gate plan for one product area. It contains a risk register, a coverage matrix, an automation and pipeline map, a contract test plan, gate criteria, quality indicators and an ownership schedule. It is used to brief the delivery sponsor and to agree test responsibilities with team leads.

Frequently Asked Questions

Is API testing strategy suitable for managers who do not write tests?

Yes. API testing strategy is set at managerial level and assumes responsibility for test planning or release decisions, not experience of writing automated tests or of a particular tool.



Which standard and models does API testing strategy rest on?

API testing strategy rests on the ISO/IEC/IEEE 29119 software testing series, the test pyramid model for test levels and Pareto analysis for defect clustering.

How is the plan from API testing strategy used at work?

You use the plan from API testing strategy to brief your delivery sponsor, set coverage and gate rules, and agree test ownership with team leads.

How does API testing strategy differ from a tool-based API testing class?

API testing strategy centres on risk, coverage, automation choices and release decisions that apply to any tool, while a tool class centres on writing checks in one product.