



Software Performance Engineering and Load Testing Course: Workload Models and Capacity Planning

31 May – 4 June 2027

London - Central London four-star



Software Performance Engineering and Load Testing Course: Workload Models and Capacity Planning

Ref: 2088_30391 **Date:** 31 May – 4 June 2027 **Location:** London - Central London four-star **Fees:** 25300 SAR

Introduction:

Software performance engineering and load testing, covering workload models and capacity planning, is a 1-week course for engineering units that build, test and run high-traffic digital services, and it ends with a Performance Test and Capacity Plan for a case organisation. Public service portals and banking channels that launch without realistic load evidence slow down or fail at peak demand, eroding trust and forcing emergency scaling. Nominees already write code, scripts or infrastructure configuration, and learning is by hands-on lab work with open-source load tools and trace data. CoreConcept Training Center delivers this software performance engineering and load testing course.

Course Objectives:

- Translate business transaction volumes and peak events into measurable performance requirements expressed as response time percentiles, throughput and error rates
- Build workload models from production access logs, choosing open or closed arrival models and a realistic transaction mix
- Design and script load, stress, soak, spike and breakpoint tests in JMeter, k6 and Gatling with correlated sessions and parameterised data
- Diagnose bottlenecks from APM traces, database query plans, garbage collection logs and thread pool metrics, and verify each tuning change by retesting
- Forecast capacity and headroom with Little's Law and the utilisation law, and set performance budgets as automated gates in CI/CD pipelines
- Produce a Performance Test and Capacity Plan that a delivery unit can run before a digital service launch

Target Audience:

- Performance test units responsible for designing, scripting and running load tests before releases
 - Application development units responsible for code paths, queries and caching that drive response times
 - Platform and infrastructure units responsible for sizing servers, containers, databases and autoscaling rules
 - Reliability and operations units responsible for production monitoring, trace data and peak-event readiness
 - Delivery and release units responsible for go-live decisions and pipeline quality gates on digital services
-

Course Outline:

Day 1: Performance Requirements, Workload Characterisation and Baseline

- Response Time Percentiles, Throughput and Concurrency Metric Definitions
- Performance Requirements Elicitation from Business Transaction Volumes
- Production Access Log Mining for Workload Mix Profiles
- Open Versus Closed Workload Models and Arrival Rates
- Current-State Performance Risk Baseline for a Digital Service

Day 2: Load, Stress, Soak and Spike Test Design Models

- Smoke, Average-Load and Breakpoint Test Profile Selection
- Stress and Spike Injection Profiles for Peak Events
- Soak Test Design for Memory Leaks and Resource Exhaustion
- Little's Law Checks for Virtual User and Pacing Sizing
- Test Environment Parity, Data Volumes and Load Generator Placement

Day 3: Scripting and Executing Load Tests with Open-Source Tools

- JMeter Thread Groups, Samplers, Timers and Correlation Extractors
- k6 Scenarios, Executors, Checks and Pass-Fail Thresholds
- Gatling Simulations with Injection Profiles and Assertions
- Parameterised Test Data, Session Tokens and Think Time Modelling
- Distributed Load Generation in CLI Mode and Results Aggregation

Day 4: Bottleneck Diagnosis, Tuning and Pipeline Performance Gates

- APM Transaction Traces and Distributed Span Breakdown Analysis
- Database Query Plans, Connection Pools and Index Tuning
- JVM Garbage Collection Pauses, Heap Sizing and Thread Pools
- Utilisation Law and Queueing Models for Capacity Forecasting
- CI/CD Performance Budgets and Regression Gate Thresholds

Day 5: Lab Capstone on the Performance Test and Capacity Plan

- Case Digital Service Workload Model and Peak Scenario Build
 - Case Load Test Execution and Percentile Results Analysis
 - Case Bottleneck Findings Report with Verified Tuning Actions
 - Case Capacity Forecast and Infrastructure Headroom Targets
 - Performance Test and Capacity Plan Completion and Review
-

Skills You Will Gain:

- Workload Modelling
- Load Test Scripting
- Performance Requirements Engineering
- Bottleneck Root Cause Diagnosis
- Database and Runtime Tuning
- Queueing-Based Capacity Forecasting
- Performance Budget Governance
- Test Results Interpretation

Why Attend This Course:

- Deliver a Performance Test and Capacity Plan to the service owner and the head of platform engineering before go-live
- Decide whether a digital service is ready to launch based on percentile results measured against agreed thresholds
- Avoid launch-day slowdowns, outages and emergency infrastructure spend caused by untested peak traffic
- Share reusable load scripts, workload model worksheets and bottleneck checklists with development and operations colleagues

Conclusion:

Back at work, the participant gives the service owner and the head of platform engineering a Performance Test and Capacity Plan that sets workload models, test profiles, percentile thresholds and infrastructure headroom for a digital service. Release managers use it to make the go-live decision, and platform units use its forecasts to size servers and autoscaling rules before peak events. After the first launch or seasonal peak, the unit should compare observed traffic, response times and resource use with the plan and revise its workload model.

Frequently Asked Questions FAQ:

What should participants know before a software performance engineering and load testing course?

Participants should already write application code, test scripts or infrastructure configuration and be able to read HTTP requests and basic logs. Bringing anonymised access logs or traffic figures from a service they support helps them build a realistic workload model during the labs.

How does software performance engineering and load testing differ from a general software testing course?

A general software testing course covers functional test design, defect management and automation across all test types. This course concentrates on non-functional behaviour under traffic: workload modelling, load tool scripting, bottleneck diagnosis from traces, queueing-based capacity forecasting and performance gates in delivery pipelines.

Which load test types does software performance engineering and load testing rely on?

Smoke, average-load, stress, spike, soak and breakpoint tests each answer a different question. Average-load confirms expected behaviour, stress and spike probe peaks, soak exposes leaks over long runs, and breakpoint finds the capacity limit where errors or response times become unacceptable.

What does a participant take back from software performance engineering and load testing training?

Each participant takes back a Performance Test and Capacity Plan for a case digital service, with a workload model, scripted test profiles, percentile thresholds, bottleneck findings and capacity forecasts, plus reusable scripts and worksheets that can be adapted to services in their own organisation.
