Performance, Load and Stress Testing Training

Duration: **2 Day**

BILGINÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Performance, Load and Stress Testing Training is designed for technical teams looking to establish performance testing practices from the beginning or incorporate them into existing software development and testing workflows. The course focuses on building the methodological understanding, infrastructure knowledge, and practical tool skills required to measure, evaluate, and improve application performance.

The program introduces essential performance testing concepts, different test types, key performance metrics, and the performance testing lifecycle. Participants will examine how performance testing relates to the Software Development Life Cycle (SDLC) and Software Testing Life Cycle (STLC), while exploring test architecture approaches and performance testing scenarios using JMeter.

The training also covers open and closed workload models, the main components of JMeter, test infrastructure preparation, and real-time monitoring of performance metrics.

By connecting performance testing principles with practical measurement and analysis, the course provides teams with a structured approach to evaluating system behavior, detecting performance bottlenecks, and improving software quality based on measurable results.

Who Should Attend

- Test engineers, quality assurance professionals, and business analysts
- Systems analysts, programmers, and software specialists
- Project managers
- Enterprise architecture professionals
- Software architects

Performance, Load and Stress Testing Training

Duration: **2 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

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

- Understand the fundamentals and objectives of performance, load, and stress testing.
- Identify different performance testing types and their use cases.
- Analyze performance metrics and evaluate test results.
- Understand the Performance Testing Life Cycle (PTLC) and its relationship with SDLC and STLC.
- Create performance testing scenarios using JMeter.
- Understand open and closed workload models.
- Explore JMeter components and testing architecture.
- Set up a testing environment and monitor real-time performance metrics.
- Identify performance bottlenecks and support system improvement efforts.

Outline

1. Performance and Performance Testing Fundamentals

- Defining performance in software systems
- Understanding performance testing and its purpose

2. Objectives and Benefits of Performance Testing

- Main goals of performance testing
- How performance testing contributes to software quality and development processes

3. Types of Performance Testing

- Overview of performance testing approaches
- Exploring different test types, including load and stress testing

4. Performance Testing Metrics

- Key metrics used to evaluate software performance
- Interpreting test results through performance measurements

5. Performance Testing Lifecycle

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Introduction to the Performance Testing Life Cycle (PTLC)
- Understanding the main stages of performance testing

6. Relationships Between SDLC, STLC, and PTLC

- Software Development Life Cycle (SDLC)
- Software Testing Life Cycle (STLC)
- Performance Testing Life Cycle (PTLC)
- Aligning performance testing with development and testing activities

7. Introduction to JMeter

- Understanding the purpose of JMeter in performance testing
- Overview of performance testing workflows using JMeter

8. Open and Closed Workload Models

- Understanding the Open Workload Model
- Understanding the Closed Workload Model
- Applying workload models to performance testing scenarios

9. JMeter Components and Architecture

- Exploring the core components of JMeter
- Understanding JMeter's architecture and operating structure

10. Live Metrics Monitoring and Test Infrastructure with JMeter

- Preparing the performance testing infrastructure
- Monitoring live performance metrics with JMeter
- Evaluating performance data collected during test execution