

# Introduction

Parasoft Virtualize simulates the behavior of systems that are still evolving, hard to access, or difficult to configure for development or testing.

Test environment access constraints have become a significant barrier to delivering quality software efficiently

- Agile and parallel development teams inevitably reach deadlocks as different teams are simultaneously working on interconnected system components—and each team needs to access the others' components in order to complete its own development and testing tasks.
- Performance test engineers need to test vs. realistic transaction performance from dependent applications (3rd party services, mainframes, packaged apps, etc.), but it's often unfeasible to set up such conditions.
- End-to-end functional testing is stymied by the difficulty of accessing all of the necessary dependencies—with the configurations you need to test against—at a single, convenient, and lengthy enough time.

Parasoft Virtualize's service virtualization provides access to the dependencies that are beyond your control, still evolving, or too complex to configure in a virtual test lab. For example, this might include third-party services (credit check, payment processing, etc.), mainframes and SAP or other ERPs. With service virtualization, you don't have to virtualize an entire system when you need to access only a fraction of its available functionality. As you naturally exercise the application under test, Parasoft captures interactions with dependencies and converts this behavior into flexible "virtual assets" with easily-configurable response parameters (e.g., performance, test data and response logic). Sophisticated virtual assets can be created and provisioned for role-based access in a matter of minutes.

With Parasoft Virtualize, you and your team can:

- Start testing whenever you're ready.
- Rapidly configure the environment conditions critical to your test plan.
- Complete the desired breadth and volume of tests.
- Confidently promote the application under test to the next level.

## End-to-end Functional Testing

Rapidly constructs test scenarios that continuously validate all critical aspects of complex transactions—which may extend through web interfaces, backend services, ESBs, databases, and everything in between.

## Security Testing

Prevents security vulnerabilities through penetration testing and execution of complex authentication, encryption, and access control test scenarios.

## Change Management

Alert appropriate team members about changes that impact their testing, enabling fast, intelligent updating of assets through a centralized change template.

## Load/Performance Testing

Verifies application performance and functionality under heavy load. Existing end-to-end functional tests are leveraged for load testing, removing the barrier to comprehensive and continuous performance monitoring. Support is also provided for load testing non-Parasoft components such as JUnits or lightweight socket-based components, and for detecting concurrency issues.

### **Message/Protocol Testing**

Automates the testing of multiple messaging and transport protocols– including HTTP, SOAP /REST, PoX, WCF, JMS, TIBCO, MQ, EJB, JDBC, RMI, and so on.

### **Event Monitoring**

During test execution, you can visualize and trace the intra-process events triggered by tests, facilitating rapid diagnosis of problems directly from the test environment. You can also continuously validate whether critical events continue to satisfy functional expectations as the system evolves.