

SilverMark's Test Mentor – Java Edition

Product Brief

Problem Statement

Modern development methodologies recognize that the earlier testing is incorporated into the development process, the lower the cost impact of defects, the shorter the project cycle and the greater the return on investment for testing.

Unit and integration (early) testing is traditionally a developer task – one which most developers appreciate the value of, but often place at the bottom of their priority lists. Testers have the mindset and the mission to create automated tests but lack the skills and the tools to create early tests on Java components. Testers end up waiting until late in the cycle to flex their testing muscles, leaving the burden of early testing on developers.

The Test Mentor Solution

The vision behind Test Mentor – *Java Edition* is to make it easy and cost effective for your team to incorporate early testing, and promote collaboration between developers and testers.

Test Mentor – *Java Edition* is an automated Java component testing solution that serves as a bridge between testers and developers, so developers and testers can collaborate on component testing, using test representations that they are comfortable with.

Java components are your business objects, which may consist of anything from a single class to an entire framework. These objects implement rules according to developers' *interpretation* of the product requirements, which may not always be correct.

Test Mentor's approach to testing components is to exercise component functionality in the same way that components are intended to be used. Components whose services are tested this way are stable and present few surprises.

Generate tests automatically

Test Mentor automates test creation by providing guided wizards and by automatically generating tests from:

- design models containing dynamic information such as sequence diagrams and state transition diagrams, such as those you might create with Rational Rose®
- Java classes, code and relationships

Test Mentor generates scalable and reusable tests that are automatically structured according to common component test design patterns.

Test representation

The tests you generate or manually create may be represented either as Java code, or as visually defined components. You can pick and choose your representation, depending on your own needs and preferences, and you can mix the two representations together.

Test Mentor's visual test definition feature **enables testers to create tests without knowing how to program in Java**. For non-developers the visual test representation hides Java complexities, such as type casting, primitive/reference type conversions, private member access, exception handling, file access, and other Java syntax and language specifics that require developer skills. At the same time, developers and code-savvy testers are free to represent tests or parts of tests as Java code.

Other features

Coverage profiling – Test Mentor profiles method coverage metrics in order to help validate the effectiveness of your tests.

Data-driven testing – Test Mentor can automatically make any test data driven. Test Mentor's built-in ability to iterate over test data files makes it easy to perform broad variation testing without writing file access code.

Tools integration – Test Mentor integrates with tools like Rational Rose and VisualAge for Java to make it easier to incorporate testing into the flow of development.

More information and contact

Web: www.javatesting.com

Information: info@javatesting.com

Sales: 888 588-0668, sales@javatesting.com



Copyright 2001 SilverMark, Inc. All Rights reserved

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. VisualAge is a trademark of IBM. Rose is a trademark of Rational Software Corporation.