



QF-TEST

Professionelles GUI-Testen für Java und Web

QF-TEST Evaluationsbericht

**Evaluation von GUI-Testtools für Java
Swing Anwendungen**

von Klaus Berg, Senior Engineer,
Siemens Corporate Technology, München
Juli 2009

Wenn es um das automatisierte Testen von Java Swing GUIs geht, spielt QF-Test definitiv in der ersten Liga. Für die Evaluierung von Tools zum Testen einer komplexen Java Web Start Anwendung hatte ich insgesamt 37 unterschiedliche Kriterien aufgestellt - priorisiert von 1 (unabdingbar) bis 3 („nice to have“), so z.B.

- **Capture & Replay Möglichkeit (auch für mehrere Bildschirme)**
- **Intelligente GUI-Komponentenerkennung**
- **Vielfalt bei den angebotenen Überprüfungsmöglichkeiten, wie Image Check oder einfacher Zugriff auf Namen von Baumknoten**
- **Konzept zur Behandlung von Fehlern in Testfällen sowie Exception/Recovery-Handling**
- **Unterstützung des sogenannten Data-Driven-Testing, d.h. Testdaten aus Tabellen, Excel/CSV-Dateien oder Datenbanken**
- **Anbindung an populäre Scriptsprachen im Java Umfeld wie Jython oder Groovy**
- **Aufbau, Nutzung und Organisation von parametrisierbaren Prozeduren (auch als Basis für „keyword-driven testing“)**
- **Integration in Ant/Maven und/oder CruiseControl**
- **Ausgabe der Testergebnisse in Form von XML und HTML**
- **Unterstützung (komplexer) Drag&Drop-Operationen (z.B. Baumknoten in Tabellenzeile)**
- **Unterstützung für das Testen von sogenannten Custom Components (GUI-Komponenten, die nicht direkt auf dem Swing-Framework aufbauen)**
- **Lizenz- und Wartungskosten**
- **Qualität des Online-Supports (sowohl technisch als auch im Hinblick auf Zeitnähe zur Anfrage)**
- **Vorhandensein von „Online-Ressourcen“ wie aktive Nutzer-Foren**
- **Qualität der Tool-Dokumentation**
- ...

Angewendet wurden diese Kriterien im Rahmen von „Basistests“ auf unterschiedliche frei verfügbare oder selbst implementierte Swing Targets (Systems Under Tests). Doch der wichtigste Prüfstein überhaupt war die Erstellung und zuverlässige Durchführung zahlreicher, z.T. schwieriger Testschritte mit dem eigentlichen Target, als „Proof-of-Concept“ für eine mögliche Kaufempfehlung.

QF-Test hat sich dabei mit über 90% Kriterien-Erfüllung gegen Open Source und andere kommerzielle Kandidaten durchgesetzt. Um es im Eiskunstlauf-Jargon auszudrücken: QF-Test hat sowohl die „Pflicht“ als auch die „Kür“ souverän gemeistert. Insbesondere die zuverlässige Komponentenerkennung und die Möglichkeit - wenn es notwendig wird - Jython oder Groovy-Skripte nahtlos in den ansonsten Event-getriebenen Testablauf zu integrieren, lassen die Vermutung aufkommen, dass es kaum Testszenarien gibt, die sich mit diesem Tool nicht realisieren lassen.

Auch die Stabilität von QF-Test ist - verglichen mit Konkurrenten - ein hervorstechendes Merkmal: kein einziger Programmabsturz und keine „Hänger“. Abrunden tut dieses äußerst positive Gesamtbild der - wie es ein anderer Kunde ausdrückt - schon „legendäre“ Support, der seinesgleichen sucht.

Fazit: Mit QF-Test kann GUI-Testen durchaus von einer „Pflichtübung“ in Begeisterung umschlagen.