



Highway 61

Qualitätssicherung und Fehleranalyse in Entwicklungs- und Migrationsprojekten.

©.haveseen - fotolia.com



In der Weiterentwicklung bestehender COBOL-Anwendungen und in COBOL-Migrationsprojekten dominieren gegenwärtig neben den gängigen Unit-Tests vor allem die Regressionstests den Bereich der Qualitätssicherung.

Diese bestehen in der Regel aus einem Satz von Ausgangsdaten, einem Szenario und einem Satz von Ergebnisdaten. Bei der Ausführung eines Szenarios mit gleichen Ausgangsdaten auf verschiedenen Systemen oder Versionsständen müssen auch die Ergebnisdaten übereinstimmen, sofern bestimmte Abweichungen nicht durch die Weiterentwicklung beabsichtigt sind.

Die Ergebnisse von Regressionstests dienen dabei vor allem dem Nachweis der Funktionsidentität bestimmter Szenarien auf unterschiedlichen Systemen bzw. in unterschiedlichen Versionsständen. Aufgrund ihrer oft relativ einfachen Realisierbarkeit und Automatisierbarkeit sind sie gerade beim Test komplexer Softwaresysteme das Mittel der Wahl. Die Komplexität der Systeme, die getestet werden, be-

Die Ergebnisse von Regressionstests dienen dabei vor allem dem Nachweis der Funktionsidentität bestimmter Szenarien auf unterschiedlichen Systemen bzw. in unterschiedlichen Versionsständen. Aufgrund ihrer oft relativ einfachen Realisierbarkeit und Automatisierbarkeit sind sie gerade beim Test komplexer Softwaresysteme das Mittel der Wahl. Die Komplexität der Systeme, die getestet werden, be-

Highlights

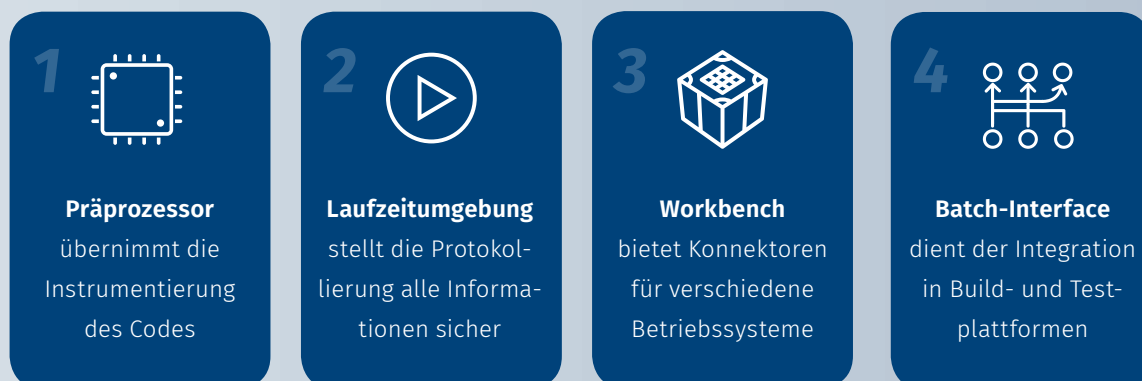
- ✓ **Reduzierung von Entwicklungs- und Debug-Zeiten** durch erleichterte Fehlersuche und den Wegfall zeitraubenden Debuggings in vielen Fällen
- ✓ **Moderne Benutzeroberfläche** mit nützlichen Werkzeugen zur Fehleranalyse
- ✓ **Regressionstests** auch bei Verwendung unterschiedlicher Systemplattformen möglich
- ✓ **Integration in bestehende Build- und Testplattformen** (ANT-Task, Kommandozeilen API, Java-API zur Integration in JUnit-Testsznarien)
- ✓ **Integration in Konzepte zur Revisionsicherheit** durch Archivierungsmöglichkeit für Szenarioläufe

dingt aber auch eine Schwäche des Testverfahrens: Es ermöglicht lediglich eine Aussage ob das System mit hoher Wahrscheinlichkeit fehlerfrei arbeitet oder nicht. Treten Fehler bzw. Abweichungen auf, sind deren Ursachen aufgrund der Komplexität des Gesamtsystems auch mit Fachkenntnissen über die internen Prozesse sehr schwer zu lokalisieren. An dieser Stelle

setzt Highway 61 an. Es ermöglicht, den Programmen im Szenariolauf „über die Schulter zu schauen“ und Abweichungen des Kontrollflusses oder bestimmter Zwischenergebnisse zu identifizieren. Dies ermöglicht in den meisten Fällen eine starke Eingrenzung von Fehlerursachen und eine signifikante Einsparung von Ressourcen für die Fehlersuche und -analyse.

Das Grundprinzip von Highway 61 besteht darin, während des Programmlaufs an bestimmten Beobachtungspunkten (Points of Interest) Informationen zu protokollieren um diese gegen die korrespondierenden Informationen eines anderen Programmlaufs in einem anderen Versionsstand oder auf einem anderen System zu vergleichen.

Das Produkt verfügt über vier Komponenten, die in den Entwicklungsprozess integriert werden:



Die Hauptkomponenten von Highway 61

Der auf Basis der bewährten EasiRun Transformationstechnologien aufbauende **Präprozessor** übernimmt die Instrumentierung des COBOL-Codes. Dabei wird zusätzlicher Code zur Protokollierung von Informationen in den Original-Quellcode injiziert. Dieser Code ist unabhängig vom verwendeten COBOL-Compiler und stellt auch für unterschiedliche Compiler ein vergleichbares Ausgabeformat der Informationen sicher.

Die **Laufzeitumgebung** stellt in Verbindung mit dem injizierten Code die Protokollierung aller Informati-

onen von COBOL-Anwendungen sicher. Die Laufzeitumgebung wird als Quellcode ausgeliefert und kann bei Bedarf auf die Besonderheiten des Testumfelds angepasst werden.

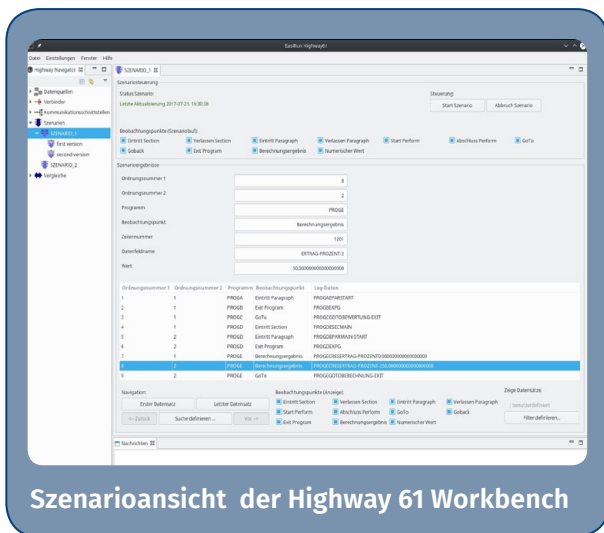
Die **Highway 61 Workbench** ermöglicht das Management und den Remote-Start von Szenarien auf verschiedenen Betriebssystemen sowie die effiziente Auswertung eines großen Volumens gewonnener Protokolldaten.



Das **Batch-Interface** ermöglicht die Integration der Highway 61-Tests in verschiedene Build- und Testplattformen, sodass Highway61 so wie andere Testframeworks auch in Continuous-Integration-Prozesse und Nightly Builds eingebunden werden kann. Zur Anbindung stehen dabei verschiedenen Schnittstellen zur Verfügung:

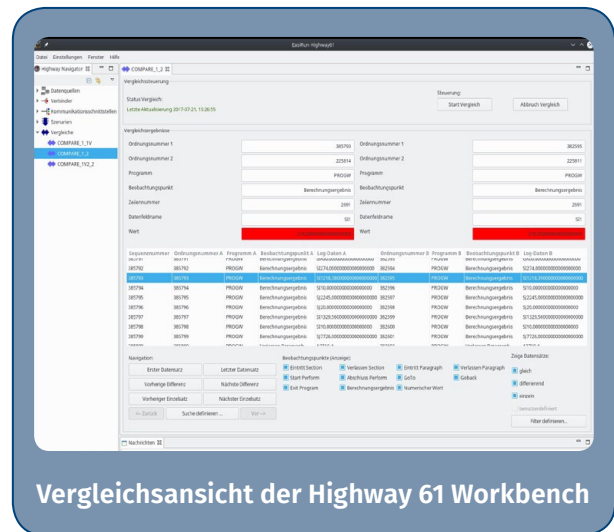
- ✓ Kommandozeilen-Interface
- ✓ Ant-Task
- ✓ Java API (z.B. Einbindung über JUnit-Framework)

Zurzeit ist Highway 61 für alle LUW-Systeme (Linux, UNIX, Windows) verfügbar. Konnektoren für verschiedene Großrechnersysteme (z/OS, z/VSE, BS2000) und andere proprietäre Systemen können bei Bedarf bereitgestellt werden. Highway 61 bietet die Möglichkeit, Protokolldaten eines Szenariolaufs zu archivieren und später gegen die Ergebnisse aktueller Szenarioläufe mit verändertem Code zu vergleichen.



Szenarioansicht der Highway 61 Workbench

Mit Hilfe dieser Funktion können verschiedene Nachweise erbracht werden. Einerseits kann belegt werden, dass der Kontrollfluss eines Szenarios durch eine Änderung nicht beeinflusst wird und andererseits, dass sich der Kontrollfluss wie erwartet geändert hat. Die archivierten Ergebnisse können als Nachweis im Rahmen der Revision verwendet werden.



Vergleichsansicht der Highway 61 Workbench

Highway 61 ist für folgende Haupteinsatzbereiche konzipiert:

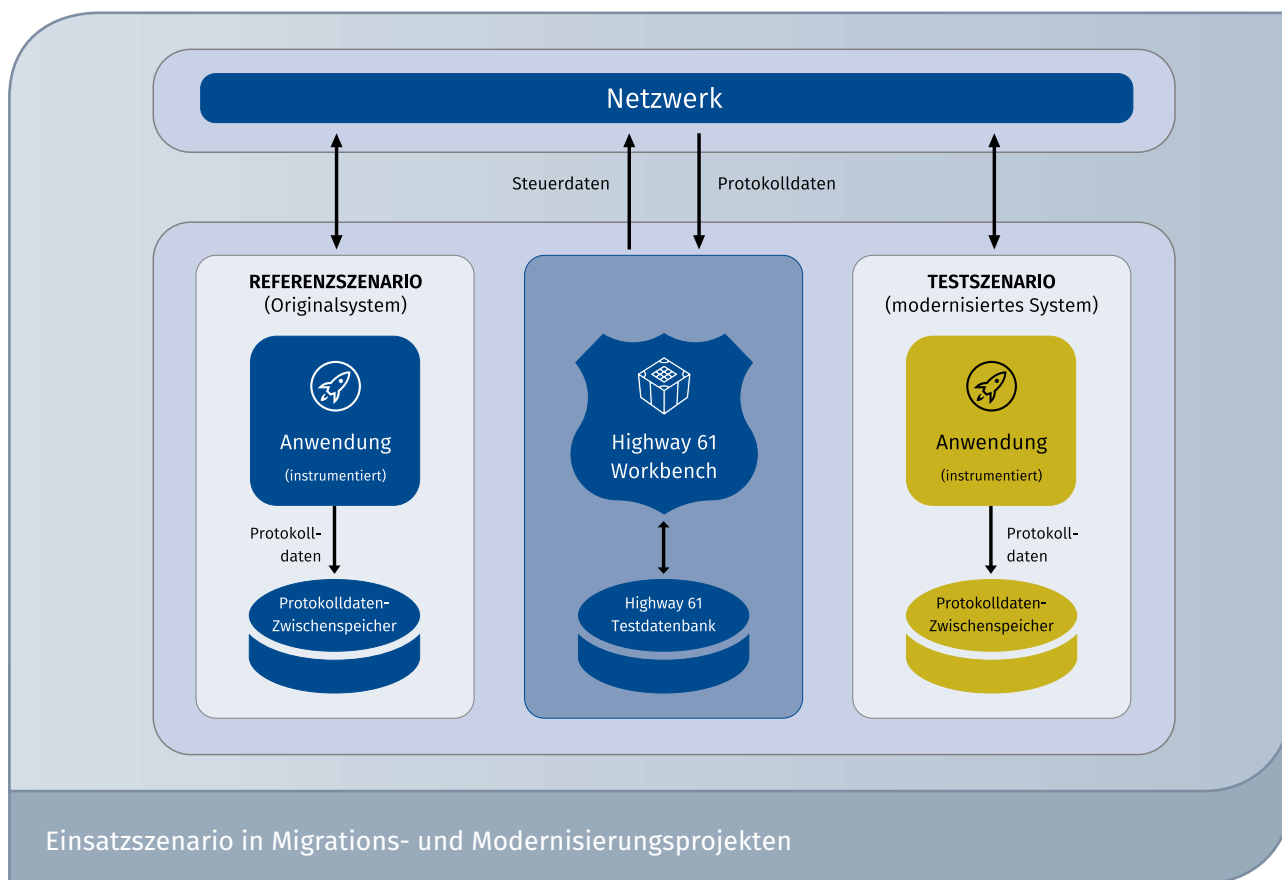
- ✓ Fortentwicklung bestehender Anwendungen
- ✓ Migrations- und Modernisierungsprojekte

Im Rahmen der Fortentwicklung von Anwendungen

können Änderungen im Programmfluss oder bestimmter Zwischenergebnisse genau nachvollzogen und dokumentiert werden (Revisionssicherheit). Bei unerwarteten Änderungen können mit Hilfe der Highway 61-Daten sehr schnell Rückschlüsse auf mögliche Fehlerursachen gezogen werden. Dies reduziert in den meisten Fällen die Aufwände für die Fehlersuche signifikant.

Beim Einsatz von Highway 61 im Rahmen von Migrations- und Modernisierungsprojekten

besteht die Möglichkeit, das Verhalten der Anwendung auf dem Legacy-System mit dem des modernisierten Systems direkt zu vergleichen. Hier kommt die Tatsache zum Tragen, dass die Protokollierung der Log-Daten system-, compiler- und plattformunabhängig erfolgt und somit eine direkte Vergleichbarkeit sichergestellt ist. Auch in diesem Einsatzszenario kann von einer signifikanten Reduzierung der Aufwände für die Fehlerlokalisierung ausgegangen werden.



Über EasiRun

EasiRun ist ein europaweit agierender Anbieter von Werkzeugen für die Software-Entwicklungsmodernisierung. In den Bereichen Mainframe, Java, .NET und COBOL (Linux, UNIX, Windows) zielt das Unternehmen mit Eclipse-IDEs, COBOL-Cross-Compilern, Datenhaltungs- und Webbrowser-Lösungen auf ganz spezielle Nischen und ermöglicht damit, komplexe IT-Modernisierungen umfassend und durchgängig erfolgreich zu machen. Zur Sicherung der Zukunftsfähigkeit und Wiederverwendbarkeit von Legacy-Applikationen entwickelt, vertreibt und implementiert EasiRun bewährte Tools mit dem Ziel, IT-Landschaften modern, unabhängig und kostengünstig zu gestalten. Das Beratungsspektrum umfasst die gesamte Anwendungs- und Anwendungsentwicklungsmodernisierung.



Informationen zu Highway 61 sowie weiteren Produkten und Lösungen von EasiRun erhalten Sie auch unter highway61.easirun.de

© Copyright 2014-2016 EasiRun Europa GmbH. Alle Rechte vorbehalten..NET, Microsoft, Windows und das Windows-Logo sind Marken der Microsoft Corp. Titelfoto © haveseen - fotolia.com/Jens Schumacher - www.idiom.io.

Alle anderen Produkt- und Firmennamen sind Marken der jeweiligen Inhaber. Alle hierin enthaltenen Materialien sind als allgemeine Information gedacht und unterliegen Änderungen. Produktbeschreibungen sind in der entsprechenden technischen Dokumentation enthalten.