

**Bearbeiter:** Stefan Löwe, stefan.loewe@tu-cottbus.de

**Datum:** 25.05.2004

**Toolname:** Junit (Version 3.8.1)

**Herkunft:** ursprünglich Erich Gamma and Kent Beck

**Zielsprachen:** Java. Es existieren aber auch Derivate für andere Sprachen (CUnit, CCUnit).

**Plattform:** plattformübergreifend (Java)

**Lizenzstatus:** IBM Public License (Open Source)

### **Kurzbeschreibung der Funktionalität:**

JUnit ist ein Unit-Test-Tool, dient also dem Testen von isolierten Programmeinheiten wie einzelne Methoden, Klassen oder Modulen. Es setzt an den öffentlichen Schnittstellen dieser Einheiten an. Trotzdem ist es ein White-Box-Testverfahren, weil es die Möglichkeit und auch die Bereitschaft voraussetzt, bestehenden Code zu ändern, insbesondere um Schnittstellen zu ändern bzw. diese überhaupt erst einzuführen um eine bestimmte Testbarkeit zu erreichen.

Mit JUnit werden wiederholbare Testfälle, also Testdaten und das erwartete Ergebnis, in derselben Programmiersprache erstellt wie der Testling selbst, hier also in Java. Diverse gesammelte Tests können als Suiten zusammengestellt werden.

Grenzen von JUnit sind Tests graphischer Benutzeroberflächen, nebenläufiger Code und mehrschichtige Architekturen. Mit Test Patterns können Teilbereiche dieser dennoch abgedeckt werden.

Der Erfolg oder Nicht-Erfolg der Tests wird farbig durch einen Balken (grün oder rot) dargestellt, der selbst in seiner Länge den Testfortschritt darstellt. Bei Nicht-Erfolg wird auf den entsprechenden Testfall verwiesen.

### **Erfahrungen im Umgang:**

- **Installation:** einfache Installation, entpacken eines ZIP-Archivs, Aufnahme in den CLASSPATH von Java
- **Stabilität:** sehr hohe Stabilität, die hohe Versionsnummer (3.8.1) und der Zeitpunkt der letzten Versionsänderung (September 2002) sprechen in diesem Fall auch dafür

- **Performanz:** durch vollständig automatischen Ablauf ohne zeitaufwendige Benutzerinteraktion sehr gut, im Wesentlichen nur abhängig vom Testling und dessen Performanz.
- **Handhabbarkeit:** durch Umsetzung in derselben Programmiersprache wie der Testling sehr gut.
- **Einarbeitungsaufwand:** die Einarbeitung in das Tool selbst erfolgt sehr schnell. Es braucht allerdings einige Zeit, bis der Anwender die zugrunde liegenden Konzepte (Test First, Test Driven Design) umsetzt.
- **Oberfläche:** sowohl textbasiert als auch graphisch (AWT oder Swing)

**Quellen:**

- <http://www.frankwestphal.de/UnitTestingmitJUnit.html>
- <http://sourceforge.net/projects/junit/>
- <http://www.junit.org/index.htm>
- „Testfieber“, c't 13/2003