

Open-Source-Programmierwerkzeuge



Andreas Zeller ist Professor für Softwaretechnik an der Universität des Saarlandes in Saarbrücken; er forscht in den Bereichen automatische Fehlersuche und Programmanalyse.

Jens Krinke ist Junior-Professor an der FernUniversität Hagen; er ist Experte für Programmanalyse und Software-Reengineering. Sein Forschungsgebiet ist die Analyse sicherheitsrelevanter Software.

dpunkt.lehrbuch

Bücher und Teachware für die moderne Informatikausbildung

Berater für die dpunkt.lehrbücher sind:

Prof. Dr. Gerti Kappel, E-Mail: gerti@big.tuwien.ac.at

Prof. Dr. Ralf Steinmetz, E-Mail: Ralf.Steinmetz@kom.tu-darmstadt.de

Prof. Dr. Martina Zitterbart, E-Mail: zit@telematik.informatik.uni-karlsruhe.de

Andreas Zeller · Jens Krinke

Open-Source- Programmierwerkzeuge

**Versionskontrolle – Konstruktion –
Testen – Fehlersuche**

2., überarbeitete und aktualisierte Auflage

unter Mitarbeit von Thomas Zimmermann



dpunkt.verlag

Andreas Zeller · Jens Krinke
autoren@programmierwerkzeuge.de

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, Düsseldorf
Druck und Bindung: Koninklijke Wöhrmann B.V., Zutphen, Niederlande

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

ISBN 3-89864-226-7

2., überarbeitete und aktualisierte Auflage 2004
Copyright © 2004 dpunkt.verlag GmbH
Ringstraße 19 B
D-69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort

Programmieren ist schwer. Noch schwerwiegender sind aber die Probleme, die im *Umfeld* des Programmierens auftreten – Probleme wie:

- ❑ Wie kann ich mein Projekt so organisieren, dass mehrere Entwickler störungsfrei arbeiten können?
- ❑ Was muss ich tun, um das Bauen meines Programms zu automatisieren?
- ❑ Wie muss ich mein Programm gestalten, dass es auf zahlreichen Maschinen läuft?
- ❑ Ich muss schnell einen Prototyp für meinen Kunden erstellen. Wie?
- ❑ Wie kann ich mein Programm automatisch testen und effizient Fehler finden und beheben?
- ❑ Wie steigere ich die Leistung meines Programms?

In Kursen oder Lehrbüchern der Programmierung tauchen solche Themen kaum auf. Hier kommt dieses Buch zu Hilfe: Wir stellen *Programmierwerkzeuge* vor, die Probleme im Umfeld des Programmierens lösen – Werkzeuge zur Versionskontrolle, zur Programmkonstruktion und -konfiguration, zur Dokumentation, zum automatischen Testen und zur Programmanalyse –, kurzum Werkzeuge für Probleme, wie sie jeden Tag in der professionellen Softwareentwicklung auftreten.

Als Werkzeuge haben wir vorwiegend »Klassiker« aus dem UNIX- und LINUX-Bereich gewählt. Für jedes Werkzeug

- ❑ stellen wir das *spezifische Problemumfeld* vor,
- ❑ zeigen wir, wie das Werkzeug *eingesetzt wird*, und
- ❑ beschreiben die *technischen Grundlagen*.

Der letzte Punkt – wie funktioniert das Werkzeug eigentlich? – hat maßgeblich unsere Auswahl bestimmt: Der *Quellcode* fast aller Werkzeuge ist frei

verfügbar, so dass jeder die Arbeitsweise nachvollziehen kann, und in der *Bedienung über Kommandozeilen* kann man die reine Grundfunktionalität ohne Schminke betrachten. Soweit vorhanden, stellen wir aber auch *grafische Benutzeroberflächen* vor; das in Kapitel 20 behandelte ECLIPSE etwa integriert fast alle hier besprochenen Werkzeuge in einer Programmierumgebung. Das Buch eignet sich damit

- ❑ für *Studenten der Informatik* und andere angehende Programmierer, die den Umgang mit typischen Programmierwerkzeugen erlernen möchten,
- ❑ für *Programmierer im UNIX- und LINUX-Umfeld*, die die wichtigsten Werkzeuge ihres Betriebssystems kennen lernen möchten, sowie
- ❑ für *fortgeschrittene Softwareentwickler*, die sich in die technischen Grundlagen ihrer Werkzeuge einarbeiten möchten.

In dieser zweiten Auflage haben wir die bestehenden Beschreibungen überarbeitet und neue Werkzeuge hinzugefügt: ANT, ANTLR, BUGZILLA, CHECKSTYLE, CPPUNIT, DOXYGEN, ECLIPSE, ELECTRIC FENCE, JAVADOC, JUNIT, VALGRIND sind jetzt alle mit eigenen Abschnitten und Kapiteln vertreten.

Wir gehen davon aus, dass Sie bereits programmieren können; Sie sollten sicher mit Ihrem Texteditor und Ihrer Programmierumgebung umgehen können. Wir wünschen uns aber, dass Sie nach dem Lesen dieses Buches auch weitere Werkzeuge gezielt einsetzen können – um *sicherer, besser* und *effizienter* zu arbeiten.

In diesem Sinne: viel Spaß beim Lesen!

Saarbrücken und Hagen, im Herbst 2003

Andreas Zeller
Jens Krinke