

# Vorwort

## Über Eclipse

Der Ursprung von Eclipse liegt in *VisualAge*, IBMs Produktlinie von integrierten Entwicklungsumgebungen für eine Vielzahl verschiedener Programmiersprachen, darunter C, C++, Fortran, Java und Smalltalk. Anfänglich waren diese Programme in der Sprache Smalltalk geschrieben, die sich Mitte der 90er Jahre großer Beliebtheit erfreute. Durch das aufkeimende frei beziehbare Java verblasste die Popularität von Smalltalk jedoch recht schnell. So erblickte im Jahre 1999 *VisualAge Micro Edition* (VAME) das Licht der Welt, das als erstes Produkt der Reihe komplett auf Java aufsetzte. VAME selbst zielte auf Java-fähige eingebettete Systeme ab.

*VisualAge*

Für seine Oberfläche nutzte VAME ein eigenes Toolkit, das *Standard Widget Toolkit* (SWT), was für Java-Applikationen sehr ungewöhnlich war, aber den Vorteil hatte, das native Feeling des Betriebssystems zu wahren. Ein flinker inkrementeller Java-Compiler wurde hinzugefügt und machte es möglich, dass Projekte zu jedem Zeitpunkt auch in übersetzter Form vorlagen. Unter der Haube war die IDE sehr flexibel, so konnte sie durch Plugins erweitert werden. Der richtige Erfolg wollte sich jedoch noch nicht einstellen.

1999: VAME

Die Erfahrungen, die die Entwickler in der *VisualAge*-Reihe sammelten, und das Framework von VAME flossen zusammen und mündeten schließlich in einem neuen Produkt, das IBM unter dem Namen *Eclipse* im November 2001 in die Öffentlichkeit trug. Die Ankündigung sorgte für viel Furore, denn im Gegensatz zu *VisualAge* wurden die Quellen offengelegt. IBM brüstete sich mit einer Wohltat an die Open-Source-Gemeinde: Der Wert der Quellen betrage 40 Millionen US-Dollar. Der Name war zynisch gewählt, denn Eclipse (zu Deutsch: Finsternis) stand in direkter Konkurrenz zu *NetBeans*, das ein Jahr zuvor durch Java-Mutter Sun (zu Deutsch: Sonne) als Open Source zur Welt gebracht wurde und ebenfalls als IDE für Java um die Gunst der Entwickler buhlte.

2001: Eclipse

*Eine Plattform für alles  
und nichts im  
Besonderen*

Eclipse ließ sich jedoch nicht auf eine Java-IDE reduzieren. Das aus VAME vererbte, jedoch verbesserte Plugin-Konzept ebnete den Weg einer universellen Entwicklungsplattform. »The Eclipse Platform is an IDE for anything, and for nothing in particular«, hieß es zeitweilig auf <http://www.eclipse.org>.

*Das Eclipse-Konsortium*

Zeitgleich zur ersten Veröffentlichung wurde das Eclipse-Konsortium *eclipse.org* von den Firmen Borland, IBM, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft und Webgain gegründet. Fortan fasste dieses Konsortium die Entwicklung der Plattform unter einem Dach zusammen. Es schrieb sich auf die Fahne, alle beigesteuerten Produkte und deren Quellen unter die von der *Open Source Initiative* [18] akzeptierte *Common Public License* (CPL) zu stellen. Zahlreiche andere Hersteller schlossen sich mit der Zeit dem Konsortium an. Die Mitglieder verpflichteten sich neben der Leistung von Beiträgen auch zur Bereitstellung von Mitarbeitern.

*2004: Eclipse wird  
gemeinnützig.*

Auf der ersten, von da an jährlich stattfindenden Eclipse-Konferenz Anfang 2004 [9] wurde bekannt gegeben, dass das Konsortium zu einer unabhängigen Non-Profit-Organisation, der *Eclipse Foundation*, umorganisiert werden sollte. Man stellte die bis dato initiierten Projekte unter die *Eclipse Public License* (EPL), und die ohnehin schon große Gemeinde bekam weitere neue Mitglieder. Inzwischen zählt sie knapp 220 Unterstützer (Stand Juni 2015).

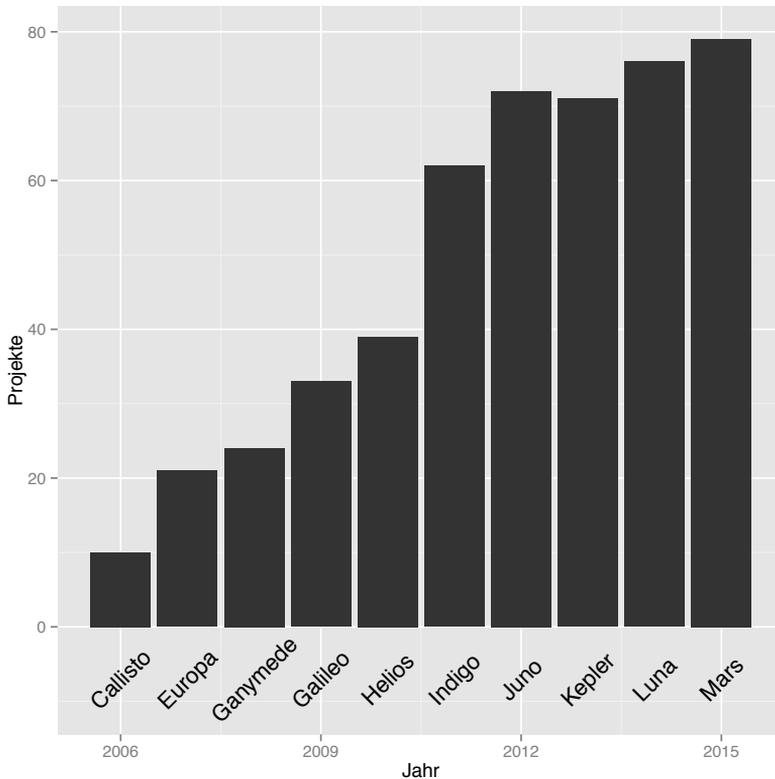
*Ab 2006: Koordinierte  
Veröffentlichung von  
Eclipse-Projekten*

Die Veröffentlichung von Eclipse findet zusammen mit weiteren Projekten wie der Java-Entwicklungsumgebung oder aber auch CDT seit dem Jahr 2006 synchronisiert statt. Die Zusammenfassung der sonst unabhängigen Projekte wurde vorgenommen, weil die Plattform Eclipse nicht nur stetige Erweiterungen erfuhr, sondern auch tiefgreifende Änderungen im Kern umfasste, die in Teilen Inkompatibilitäten zu vorherigen Revisionen verursachten. Benutzer können zumindest von den unter gleicher Bezeichnung veröffentlichten Paketen sicher sein, dass sie ohne solche auftretenden Kompatibilitätsprobleme miteinander harmonieren. In der Regel folgen einer koordinierten Veröffentlichung innerhalb eines Jahres zwei sogenannte Service-Releases, die fast ausschließlich Bugfixes beinhalten, zumindest was die Eclipse-Plattform angeht, bevor im Sommer des nächsten Jahres der neue Zyklus beginnt.

*Jupitermonde als  
Namensgeber*

Die Zyklen trugen in der Vergangenheit zunächst die Namen von Jupitermonden: Callisto (2006, 10 Projekte), Europa (2007, 21 Projekte) und Ganymede (2008, 24 Projekte). Von diesem Namensschema verabschiedete man sich bereits im Jahr 2009, da man befürchtete, dass die Anzahl der Jupitermonde (von denen laut Wikipedia heute mindestens 67 bekannt sind) nicht hoch genug ist. So ließ es sich jedenfalls ein Eclipse-Entwickler in einem Interview im Eclipse-Magazin [3] entlocken:

We changed the naming scheme because we were concerned that we were going to run out of names long before we ran out of steam on the release train...



**Abb. 1**

Anzahl der Eclipse-Projekte, die bei den jährlichen Veröffentlichungen teilnahmen

Der Release-Zyklus von 2009, der 33 Projekte beinhaltete, erhielt schließlich den Namen Galileo, zu Ehren jenes italienischen Astronoms, der 400 Jahre zuvor mit einem selbst gebauten Fernrohr u. a. die bisherigen Namensgeber entdeckte. Über die folgenden Zyklen gibt Abbildung 1 Aufschluss. Hervorzuheben ist dabei das Juno-Release, das ganz im Zeichen der neuen Eclipse-4-Plattform stand, die jedoch anfänglich insbesondere aufgrund von Performance-Regressionen durch eher negative Schlagzeilen auf sich aufmerksam machte [16]. Inzwischen sind diese aber weitgehend behoben. Auch Features wie die Möglichkeit des Umsortierens von Perspektiven, die die Eclipse-Plattform vor Version 4.0 mit sich brachte, sind spätestens seit Luna wieder verfügbar, die Vorgängerversion von Mars, auf der dieses Buch basiert.

Koppelte man zu Beginn den Veröffentlichungszyklus der teilnehmenden Projekte tatsächlich fest an den Zyklus der koordinierten Veröffentlichungen, ist es inzwischen nicht unüblich, dass einzelne Projekte

*Weitere Entwicklung*

*Lockerung in der Projektveröffentlichung*

pro Eclipse-Jahr häufigere stabile Veröffentlichungen vornehmen. Bei CDT einigte man sich, bis zu zwei Zwischenversionen außerhalb des jährlichen Zyklus einzuschieben, die Teil der Service-Releases sind. Beispielsweise beinhaltet die erste Luna-Veröffentlichung im Jahr 2014 CDT 8.4, während das zweite Service-Release von Luna, das im Februar 2015 fertiggestellt wurde, CDT 8.6 beinhaltet.

## Über CDT

### *Entstehung von CDT*

Ein halbes Jahr nach der ersten Veröffentlichung von Eclipse legte Konsortiums-Gründungsmitglied und Echtzeitbetriebssystem-Spezialist QNX Teile der Quellen seiner *QNX Momentics Development Suite* offen. QNX kündigte an, die Suite in das Eclipse-Framework einzufügen, und übernahm auch die Leitung des jungen CDT-Projekts, das zum Ziel hat, Eclipse mit einer flexiblen Unterstützung der Programmiersprachen C und C++ zu ergänzen. Im April 2003 folgte dann die erste Veröffentlichung von CDT für Eclipse 2.0. So wurde die Behauptung, Eclipse sei eine Mehrsprachenplattform, das erste Mal Realität: Neben einfachem Syntax Highlighting für C und C++ integrierte man die GNU-Toolchain, wobei auch der Debugger nicht ausgelassen wurde.

### *Aktuelle Version*

CDT war auch Teil der ersten koordinierten Veröffentlichung der Eclipse-Projekte im Jahr 2006, zu der es CDT schon bis zur Version 3.1 geschafft hatte. 2010 wurde bereits Version 7.0 veröffentlicht, und seit Juni 2015 liegt CDT in Version 8.7 vor, die auch Gegenstand dieses Buchs ist. Der schnelle Sprung zu hohen Versionsnummern zeigt, dass CDT eine sehr dynamische Entwicklung hinter sich hat.

CDT macht vom Angebot der Eclipse-Plattform Gebrauch, eigene spezifische Erweiterungspunkte bereitzustellen. Das erlaubt auch für CDT ein gewisses Maß an Flexibilität. So ist es für einen Anbieter einer Compiler-Suite sehr leicht möglich, diese in das Eclipse-/CDT-Framework einzubinden.

## Zielsetzung des Buchs

Es sind bereits zahlreiche Bücher über Eclipse erschienen. Schaut man sich den Markt jedoch genauer an, so stellt man schnell fest, dass sich die meisten Werke dem Produkt Eclipse nur in Zusammenhang mit der Java-Entwicklung nähern. Andere Plugins oder andere Programmiersprachen finden, wenn überhaupt, nur am Rande Erwähnung. Dieses Buch möchte die Lücke für die Programmiersprache C und ihren Abkömmling C++ schließen und den Leser in die Benutzung von Eclipse und CDT einführen.

Kenntnisse über Eclipse sind nicht notwendig. Weiterhin ist es unerheblich, in welcher Art und Weise der Leser seine C- oder C++-Projekte bisher verwirklicht hat, sei es mittels einer Entwicklungsumgebung wie Microsofts *Visual Studio* oder ganz traditionell über Texteditor, *Make* und Kommandozeile.

*Vorkenntnisse*

In die Programmiersprachen C und C++ kann das Buch jedoch nicht einführen. Hierfür existiert bereits sehr gute und umfangreiche Literatur [14], die teilweise sogar kostenlos angeboten wird, wie z. B. im Falle des Wikibooks *C-Programmierung*, auf das unter der URL <http://de.wikibooks.org/wiki/C-Programmierung> zugegriffen werden kann. Die Handhabung von Werkzeugen, die unmittelbar mit der Programmiersprache assoziiert sind, wie z. B. *Make*, steht ebenso wenig im Vordergrund. Dennoch ist das Buch auch für C- oder C++-Einsteiger als Ergänzung zu einem Einführungsbuch von Interesse, da eine Umgebung wie CDT dem Entwickler jede Menge Hilfestellungen in den einzelnen Entwicklungsphasen liefert und es ihm erlaubt, sich auf das Wesentliche zu konzentrieren. Gerade Neulinge werden dies als praktisch empfinden. Beispielsweise sind Kenntnisse über das erwähnte *Make* bei vielen Projekten gar nicht nötig.

*Keine Einführung in C oder C++*

Das Buch beschreibt CDT in seiner rohen Form, so wie es von *eclipse.org* heruntergeladen werden kann. Erweiterungen, die in den CDT-basierten Entwicklungsumgebungen von Drittanbietern zu finden sind, spielen nur eine geringe Rolle. Da das Buch eine benutzerorientierte Einführung darstellt, wird auch die Beschreibung der API außen vor gelassen, die ausschließlich für die Entwicklung von Erweiterungen für CDT von Bedeutung ist. Ein solches Vorhaben erfordert selbstverständlich Kenntnisse über viele Konzepte und Abläufe von CDT, auf die in den entsprechenden Abschnitten detailliert eingegangen wird. Insofern werden Sie auch als potenzieller Plugin-Anbieter von der Lektüre profitieren können.

*Hauptsächlich CDT »pur«*

## Buchaufbau

Die Grobgliederung des Buchs besteht aus drei Teilen. Der erste Teil führt Sie in die Welt von Eclipse und CDT ein. Los geht's mit einer Installationsbeschreibung, die je nach verwendeter Plattform und Anforderung mehr oder weniger kompliziert durchgeführt werden muss. Anschließend werden Sie in die Begriffe und die wichtigsten Prozeduren anhand eines eingängigen Beispiels eingeführt. So können Sie bereits nach kurzer Zeit einfache C- oder C++-Projekte mit Eclipse/CDT erstellen, bearbeiten, übersetzen, ausführen und debuggen.

*Einführung*

*CDT als Unterstützer in  
den einzelnen Phasen  
des  
Entwicklungsprozesses*

Der zweite Teil orientiert sich an den einzelnen Phasen, die ein C- oder C++-Programmierer zu bewerkstelligen hat, damit am Ende ein lauffähiges Produkt zustande kommt. Es wird aufgezeigt, was mit Eclipse/CDT derzeit möglich ist, bei welchen Problemen Ihnen diese Kombination Hilfestellungen geben kann und was derzeit noch nicht oder nur unzureichend implementiert ist. Zusätzlich zu der Beschreibung der Bedienung werden konkrete Problemstellungen aufgezeigt, die unter Zuhilfenahme kleiner Beispiele eine Lösung finden.

*Weiterführende  
Themen*

Der dritte Teil widmet sich schließlich weiterführenden Themen. Sie werden z. B. erfahren, wie Sie Ihre Projekte mit Eclipse in einem Team organisieren, mit Hilfe von zusätzlichen Plugins aufgabenfokussierte Entwicklung betreiben oder Debugging-Tools wie Valgrind in den Entwicklungsprozess einbeziehen.

## Terminologie

Eclipse ist internationalisiert und kann so theoretisch in anderen Sprachen als Englisch betrieben werden. Allerdings wurde bisher nur für die Version 3.2 von Eclipse ein komplettes Sprachpaket durch IBM gesponsert, das für aktuelle Versionen nur beschränkte Gültigkeit hat. Aus diesem Grund wird hier mit der englischen Version des Programmpakets gearbeitet. Die im Buch verwendete Terminologie, z. B. bei der Beschreibung von Benutzeroberflächenelementen, ist deshalb ebenso an die englischen Fachausdrücke angelehnt, wenn dafür keine adäquate Übersetzung bekannt ist.

## Webseite und Kontakt

Zum Buch existiert eine Webseite, die Sie unter

<http://www.dpunkt.de/buecher/4680.html>

abrufen können. Neben den Quelltexten und einer Errata-Liste sind dort alle im Buch aufgeführten Verweise auf Webressourcen verzeichnet. Zusätzlich finden Sie einen Katalog von Eclipse-Erweiterungen, die besonders C- oder C++-Programmierer interessieren könnten.

Haben Sie Fragen zum Buch, einen Fehler entdeckt, Vorschläge oder Kritiken? Dann können Sie mir unter [mail@sebastianbauer.info](mailto:mail@sebastianbauer.info) eine Nachricht zukommen lassen.