Geleitwort

Lassen Sie sich von dem Titel nicht irreführen!

Was Sie hier vor sich haben, ist ein wirklich sehr, sehr gutes Buch über Designprinzipien, Programmiertechniken, testgetriebene Entwicklung und handwerkliches Können – und dann geben die ihm einen Titel wie *Modern C++ Programming with Test-Driven Development*![1] Seufz!

Verstehen Sie mich nicht falsch: In diesem Buch geht es tatsächlich um moderne C++-Programmierung. Als C++-Entwickler wird Ihnen der Code in diesem Buch gefallen. Es ist randvoll mit wirklich interessantem und gut geschriebenem C++-Code und enthält wohl mehr Code als Text. Blättern Sie mal das Buch durch und schauen Sie sich an, wie viele Seiten ohne Code es gibt. Nicht viele, möchte ich wetten. Wenn Sie also nach einem guten Buch suchen, das Ihnen die modernen Techniken von C++ beibringt, dann halten Sie schon das richtige in Ihren Händen.

In diesem Buch geht es aber um mehr als nur um moderne C++-Entwicklung – um *viel* mehr. Erstens ist es wahrscheinlich die vollständigste und verständlichste Darstellung der testgetriebenen Entwicklung (Test-Driven Development, TDD), die ich je gesehen habe (und ich habe eine Menge gesehen!). Praktisch jeder Aspekt von TDD, dem wir in den letzten anderthalb Jahrzehnten begegnet sind, wird hier behandelt, von instabilen Tests zu Mocks, von der Londoner zur Clevelander Schule, von der Regel »eine Annahme pro Test« bis zum Muster »Given-When-Then«. All das und noch viel mehr finden Sie in diesem Buch. Es ist aber keine Sammlung unzusammenhängender akademischer Texte. Im Gegenteil, in diesem Buch werden die einzelnen Aspekte anhand von Beispielen und Fallstudien durchgearbeitet. Es zeigt die Probleme und Lösungen in Form von Code.

Müssen Sie C++-Entwickler sein, um dieses Buch zu verstehen? Natürlich nicht. Der C++-Code ist so sauber und so gut geschrieben und die Prinzipien sind so deutlich, dass Sie auch als Java-, C#-, C- oder sogar Ruby-Programmierer keine Schwierigkeiten damit haben werden.

1. Der Titel der deutschen Übersetzung wurde in »Testgetriebene Entwicklung mit C++« geändert, da dieser Titel auch aus Sicht der Fachgutachter den Inhalt des Buches besser wiedergibt.

Und dann sind da die Designprinzipien. Menschenskind, dieses Buch ist ja eine Anleitung zum guten Design! Sie lernen hier Schritt für Schritt ein Prinzip nach dem anderen kennen, ein Problem nach dem anderen und eine Technik nach der anderen: vom Single-Responsibility-Prinzip zur Umkehrung von Abhängigkeiten, von der Isolierung der Schnittstelle zu den agilen Prinzipien des einfachen Designs, von DRY zu »tell, don't ask«. Dieses Buch ist eine Goldgrube an Ideen und Lösungen für das Softwaredesign. Und auch hier werden diese Ideen wiederum im Zusammenhang mit praktischen Problemen und praktischen Lösungen in echtem Code dargestellt.

Außerdem geht es auch noch um Programmiertechniken. Dieses Buch ist randvoll damit, von kleinen Methoden bis zum Pair Programming, von Katas bis zu Variablennamen. Sie finden hier nicht nur haufenweise Code, aus dem Sie gute Vorgehensweisen ableiten können – der Autor bearbeitet jeden einzelnen Punkt auch mit genau der richtigen Menge an Diskussion und ausführlicher Darstellung.

Der Titel dieses Buches ist komplett falsch. Dies ist kein Buch über C++-Programmierung, sondern über solide Software-Handwerkskunst, das nur zufällig C++ als Sprache für die Beispiele verwendet. Ein besserer Titel wäre gewesen: *Software Craftsmanship: With Examples in Modern C++*.

Wenn Sie also ein Entwickler für Java, C#, Ruby, Python, PHP, VB oder gar COBOL sind, sollten Sie dieses Buch lesen. Lassen Sie sich nicht von dem »C++« im Titel abschrecken. Lesen Sie das Buch trotzdem. Und lesen Sie dabei den Code! Er ist nicht schwer zu verstehen. Während Sie bei der Lektüre gute Designprinzipien, Programmiertechniken, handwerkliches Können und testgetriebene Entwicklung lernen, werden Sie wahrscheinlich auch feststellen, dass ein bisschen C++ niemandem weh tut.

Robert »Uncle Bob« Martin
Gründer von Object Mentor Inc.