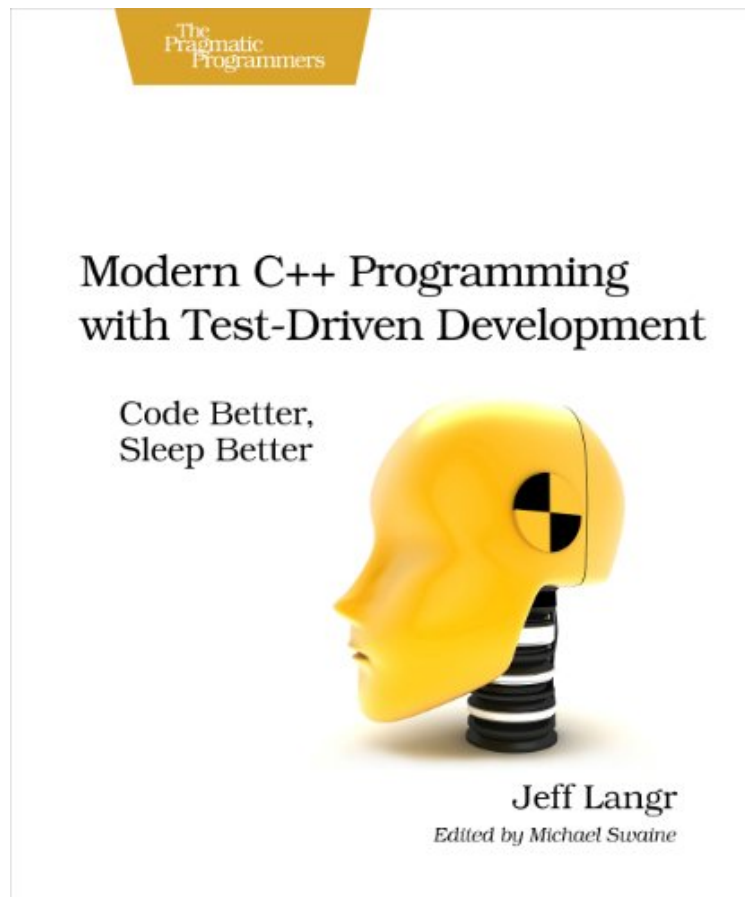


Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better

Von Jeff Langr

**Download PDF | ePub | DOC | audiobook | ebooks*



 Download

 Read Online

Produktinformation - Verkaufsrang: #166667 in eBooks Veröffentlicht am: 2013-10-10 Erscheinungsdatum: 2014-01-13 File Name: B00HUEG8M8 | File size: 39.Mb

Von Jeff Langr : Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better before purchasing it in order to gauge whether or not it would be worth my time, and all praised Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better:

Kundenrezensionen Hilfreichste Kundenrezensionen 12 von 12 Kunden fanden die folgende Rezension hilfreich. Kommt nicht zur Sache Von Roland Wolf Ich habe mich gefreut, als endlich ein Test-Driven Buch für C++ erschienen ist. Man kann natürlich eines der vielen Java-Bücher nehmen, aber in der C++ Welt sind die Tools und Frameworks doch anders. Leider blendet beim Lesen des Buchs das Licht des Offensichtlichen ziemlich oft. Die eigentlich interessanten Themen spricht der Autor gar nicht erst an. Die Beispiele die Langr nennt erwähnen nicht die Typischen Hindernisse, die in wirklichen Projekten TDD schwer machen: Singletons, Designfehler wie Objekterzeugung im Konstruktor, Untrennbarkeit der Schichten, asynchrone Programmierung, Timer. Wie integriere ich die Tests in das Build-System? Welche Seams gibt es? Wer das Buch gelesen hat wird motiviert sein TDD einzusetzen aber einfach noch viele

Wissenslücken haben. Das billige Papier und das lieblose Layout kann man bei einem Technikbuch verschmerzen. Für Anfänger ist das Buch eine gute Einführung. Für den Praktiker ist es doch ein bisschen lahm. 0 von 1 Kunden fanden die folgende Rezension hilfreich. TDD funktioniert eben nicht nur mit Skriptingsprachen. Von Philipp H. Toner. Nachdem der C++-Compiler ein valides, vollständiges Programm zu kompilieren, unabhängig von den Tests, die ich laufen lassen möchte, erschien es mir immer als schier unmöglich TDD in meine C++ Projekte einfließen zu lassen. In Skriptingsprachen wie PHP oder Ruby konnte man wesentlich leichtgewichtiger hin und her springen und Tests einbauen. Nun, es ist die agile Softwareentwicklung, die es erlaubt iterative Software zu entwickeln und das mit TDD. Nach jeder Iteration habe ich eine funktionierende getestete Softwarekomponente, die sich kompilieren und ausführen lässt. Für die Tests wird gtest/gmock verwendet und das mit vorbildlicher Hamcrest Notation, die aussagekräftige Fehlermeldungen ausgibt, nichts mit "assert that false is true" (oh Wunder). Favorit und Neuheit im TDD Kontext für mich ist der Red-Green-Ansatz, den ich inzwischen sehr intensiv via Splitscreen in meiner IDE praktiziere. 0 von 1 Kunden fanden die folgende Rezension hilfreich. Ehrlich und Vollständig. Von Andreas Pokorny. Ich denke Test Driven Development kann man nur durchs machen lernen, und genau darauf zieht das Buch ab. Schrittweise - ja genau auch das ist essentiell und wird vermittelt: die richtige Schrittgröße - list man einfache Programmieraufgaben und sieht worauf es bei TDD ankommt. Wer nicht das Glück hat durch einen Teamkollegen auf den TDD Pfad geführt zu werden muss zu diesem Buch greifen.

Kurzbeschreibung If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this book, Jeff Langr gives you hands-on lessons in the challenges and rewards of doing TDD in C++. **Modern C++ Programming With Test-Driven Development**, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++. **What You Need** A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include: - cURL- JsonCpp- Boost (filesystem, date_time/gregorian, algorithm, assign) Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp. **Pressestimmen** "Jeff Langr has written another excellent book. This time he brings Test-Driven Development to the world of C++. Jeff's starting example brings us face to face with the stark simplicity of good TDD. He explains why we work that way and then provides key practical details, all the way through test doubles, working with legacy code, threading, and more. This one is a keeper for anyone working with C++!"--Ron Jeffries, Co-creator of the Extreme Programming methodology "Jeff Langr has written the best C++ book in years. **Modern C++ Programming with Test-Driven Development** is the perfect mixture of theory and practice. The abstractions are explained with clarity and gusto, but the details are right there when you need them. It's sure to be an instant classic, in C++ and in TDD both."--Michael D. Hill, XP coach and writer "Jeff is an expert software craftsman, and in this book he shares his considerable wisdom about crafting great software. This book is not about testing, though you will learn valuable testing techniques. It is about improving your skills, code, products, and life through test-driving. Whether you're a novice, expert, or in between, Jeff will show you the how and the why of an expert who test-drives better products with C++."--James W. Grenning, Author of "Test-Driven Development for Embedded C" **Kurzbeschreibung** If you program in C++ you've been neglected. Test-driven development (TDD) is a modern software development practice that can dramatically reduce the number of defects in systems, produce more maintainable code, and give you the confidence to change your software to meet changing needs. But C++ programmers have been ignored by those promoting TDD--until now. In this book, Jeff Langr gives you hands-on

lessons in the challenges and rewards of doing TDD in C++. *Modern C++ Programming With Test-Driven Development*, the only comprehensive treatment on TDD in C++ provides you with everything you need to know about TDD, and the challenges and benefits of implementing it in your C++ systems. Its many detailed code examples take you step-by-step from TDD basics to advanced concepts. As a veteran C++ programmer, you're already writing high-quality code, and you work hard to maintain code quality. It doesn't have to be that hard. In this book, you'll learn: how to use TDD to improve legacy C++ systems how to identify and deal with troublesome system dependencies how to do dependency injection, which is particularly tricky in C++ how to use testing tools for C++ that aid TDD new C++11 features that facilitate TDD As you grow in TDD mastery, you'll discover how to keep a massive C++ system from becoming a design mess over time, as well as particular C++ trouble spots to avoid. You'll find out how to prevent your tests from being a maintenance burden and how to think in TDD without giving up your hard-won C++ skills. Finally, you'll see how to grow and sustain TDD in your team. Whether you're a complete unit-testing novice or an experienced tester, this book will lead you to mastery of test-driven development in C++.

What You Need A C++ compiler running under Windows or Linux, preferably one that supports C++11. Examples presented in the book were built under gcc 4.7.2. Google Mock 1.6 (downloadable for free; it contains Google Test as well) or an alternate C++ unit testing tool. Most examples in the book are written for Google Mock, but it isn't difficult to translate them to your tool of choice. A good programmer's editor or IDE. cmake, preferably. Of course, you can use your own preferred make too. CMakeLists.txt files are provided for each project. Examples provided were built using cmake version 2.8.9. Various freely-available third-party libraries are used as the basis for examples in the book. These include:- cURL- JsonCpp- Boost (filesystem, date_time/gregorian, algorithm, assign) Several examples use the boost headers/libraries. Only one example uses cURL and JsonCpp.