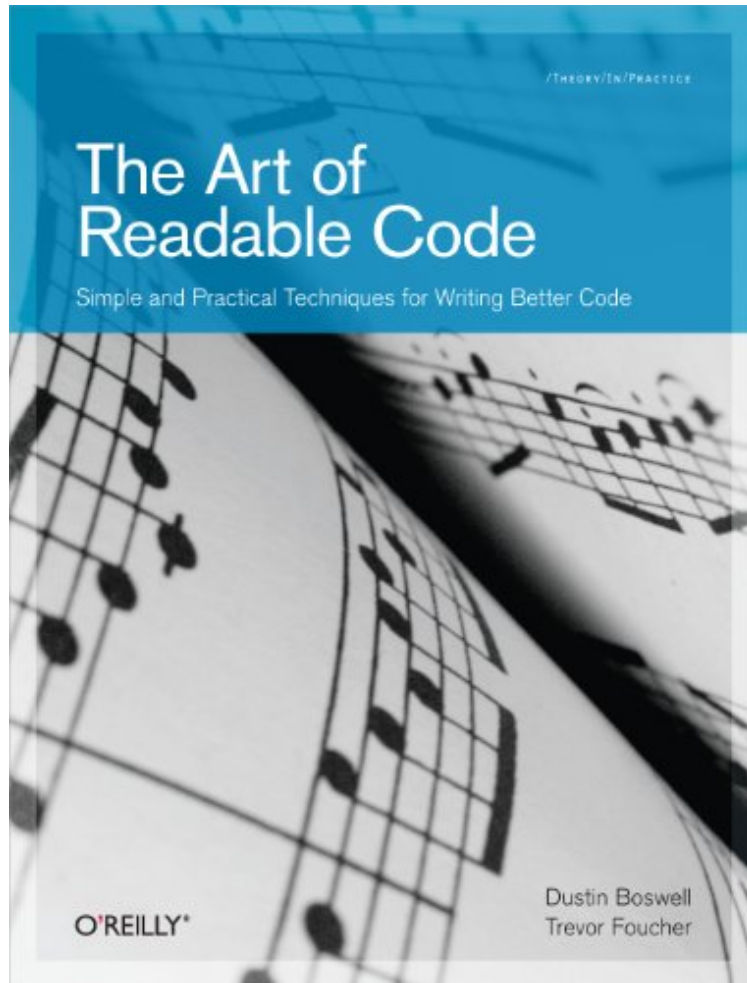


[Read download] The Art of Readable Code: Simple and Practical Techniques for Writing Better Code (Theory in Practice)

The Art of Readable Code: Simple and Practical Techniques for Writing Better Code (Theory in Practice)

Von *Dustin Boswell, Trevor Foucher*
DOC | *audiobook | ebooks | Download PDF | ePub



Produktinformation -Verkaufsrank: #341118 in eBooksVerffentlicht am: 2011-11-03Erscheinungsdatum:
2011-11-03File Name: B0064CZ1XE | File size: 64.Mb

Von Dustin Boswell, Trevor Foucher : The Art of Readable Code: Simple and Practical Techniques for Writing Better Code (Theory in Practice) before purchasing it in order to gauge whether or not it would be worth my time, and all praised The Art of Readable Code: Simple and Practical Techniques for Writing Better Code (Theory in Practice):

KundenrezensionenHilfreichste Kundenrezensionen0 von 0 Kunden fanden die folgende Rezension hilfreich. Ein Muss fr jeden ProgrammiererVon 42BSDieses Buch ist lesenswert und sollte wirklich von _jedem_ gelesen werden, der programmiert.Wenn auch nicht alle Ideen fr jedes Projekt sinnvoll sind, so kann man doch zumindest Anstze mitnehmen.1 von 1 Kunden fanden die folgende Rezension hilfreich. Guter berblick und einiges zum nachdenken,

speziell für jüngere Programmierer Von Mirko Mein Lieblingszitat aus Abelson, Sussman Sussman: Structure and Interpretation of Computer Programs: "... programs must be written for people to read, and only incidentally for machines to execute" Aber wie macht man das? The Art of Readable Code versucht das zu beantworten.- Wann ist Code lesbar?- Was macht einen guten Namen für eine Variable oder eine Funktion aus?- Was bedeutet gute Lesbarkeit?- Welchen Sinn erfüllen Kommentare? Das klingt trivial und ein guter Entwickler wird das wissen, aber dieses Buch fasst dieses implizite Wissen gut zusammen. Ein must read - insbesondere für junge Software Entwickler. 2 von 3 Kunden fanden die folgende Rezension hilfreich. Informativ und gut zu lesen Von sad rane Neben mir bereits bekannten Vorgehensweisen konnte ich einige Anregungen mitnehmen. Meine Erfahrungen beim Lesen von existierendem Code zeigen, dass zu wenige Programmierer sich mit dem Thema beschäftigt haben. Es ist relativ leicht zu lesen und die Code-Beispiele sind eine gute Illustration des Anliegen der Autoren sowie der Vorgehensweise. Auch wenn sich manche Vorschläge trivial anfühlen und man auch nicht mit allen Ansichten bereinstimmt, lohnt es sich doch, darüber nachzudenken und die eigene Arbeit kritisch zu beleuchten.

Kurzbeschreibung As programmers, we've all seen source code that's so ugly and buggy it makes our brain ache. Over the past five years, authors Dustin Boswell and Trevor Foucher have analyzed hundreds of examples of "bad code" (much of it their own) to determine why they're bad and how they could be improved. Their conclusion? You need to write code that minimizes the time it would take someone else to understand it even if that someone else is you. This book focuses on basic principles and practical techniques you can apply every time you write code. Using easy-to-digest code examples from different languages, each chapter dives into a different aspect of coding, and demonstrates how you can make your code easy to understand. Simplify naming, commenting, and formatting with tips that apply to every line of code. Refine your programs loops, logic, and variables to reduce complexity and confusion. Attack problems at the function level, such as reorganizing blocks of code to do one task at a time. Write effective test code that is thorough and concise as well as readable. Being aware of how the code you create affects those who look at it later is an important part of developing software. The authors did a great job in taking you through the different aspects of this challenge, explaining the details with instructive examples. "Michael Hunger, passionate Software Developer