



Clean Code: A Handbook of Agile Software Craftsmanship

By Robert C. Martin



Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin

Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way.

Noted software expert Robert C. Martin presents a revolutionary paradigm with *Clean Code: A Handbook of Agile Software Craftsmanship*. Martin has teamed up with his colleagues from Object Mentor to distill their best agile practice of cleaning code "on the fly" into a book that will instill within you the values of a software craftsman and make you a better programmer—but only if you work at it.

What kind of work will you be doing? You'll be reading code—lots of code. And you will be challenged to think about what's right about that code, and what's wrong with it. More importantly, you will be challenged to reassess your professional values and your commitment to your craft.

Clean Code is divided into three parts. The first describes the principles, patterns, and practices of writing clean code. The second part consists of several case studies of increasing complexity. Each case study is an exercise in cleaning up code—of transforming a code base that has some problems into one that is sound and efficient. The third part is the payoff: a single chapter containing a list of heuristics and "smells" gathered while creating the case studies. The result is a knowledge base that describes the way we think when we write, read, and clean code.

Readers will come away from this book understanding

- How to tell the difference between good and bad code
- How to write good code and how to transform bad code into good code
- How to create good names, good functions, good objects, and good classes
- How to format code for maximum readability
- How to implement complete error handling without obscuring code logic
- How to unit test and practice test-driven development

This book is a must for any developer, software engineer, project manager, team lead, or systems analyst with an interest in producing better code.

 [Download Clean Code: A Handbook of Agile Software Craftsman ...pdf](#)

 [Read Online Clean Code: A Handbook of Agile Software Craftsm ...pdf](#)

Clean Code: A Handbook of Agile Software Craftsmanship

By Robert C. Martin

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin

Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way.

Noted software expert Robert C. Martin presents a revolutionary paradigm with *Clean Code: A Handbook of Agile Software Craftsmanship*. Martin has teamed up with his colleagues from Object Mentor to distill their best agile practice of cleaning code "on the fly" into a book that will instill within you the values of a software craftsman and make you a better programmer—but only if you work at it.

What kind of work will you be doing? You'll be reading code—lots of code. And you will be challenged to think about what's right about that code, and what's wrong with it. More importantly, you will be challenged to reassess your professional values and your commitment to your craft.

Clean Code is divided into three parts. The first describes the principles, patterns, and practices of writing clean code. The second part consists of several case studies of increasing complexity. Each case study is an exercise in cleaning up code—of transforming a code base that has some problems into one that is sound and efficient. The third part is the payoff: a single chapter containing a list of heuristics and "smells" gathered while creating the case studies. The result is a knowledge base that describes the way we think when we write, read, and clean code.

Readers will come away from this book understanding

- How to tell the difference between good and bad code
- How to write good code and how to transform bad code into good code
- How to create good names, good functions, good objects, and good classes
- How to format code for maximum readability
- How to implement complete error handling without obscuring code logic
- How to unit test and practice test-driven development

This book is a must for any developer, software engineer, project manager, team lead, or systems analyst with an interest in producing better code.

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin Bibliography

- Sales Rank: #1788 in Books
- Brand: imusti
- Published on: 2008-08-11
- Original language: English
- Number of items: 1
- Dimensions: 9.10" h x 1.10" w x 6.90" l, 1.68 pounds

- Binding: Paperback
- 464 pages



[Download](#) Clean Code: A Handbook of Agile Software Craftsman ...pdf



[Read Online](#) Clean Code: A Handbook of Agile Software Craftsm ...pdf

Download and Read Free Online Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin

Editorial Review

From the Back Cover

Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way.

Noted software expert Robert C. Martin presents a revolutionary paradigm with "Clean Code: A Handbook of Agile Software Craftsmanship." Martin has teamed up with his colleagues from Object Mentor to distill their best agile practice of cleaning code "on the fly" into a book that will instill within you the values of a software craftsman and make you a better programmer--but only if you work at it.

What kind of work will you be doing? You'll be reading code--lots of code. And you will be challenged to think about what's right about that code, and what's wrong with it. More importantly, you will be challenged to reassess your professional values and your commitment to your craft.

"Clean Code" is divided into three parts. The first describes the principles, patterns, and practices of writing clean code. The second part consists of several case studies of increasing complexity. Each case study is an exercise in cleaning up code--of transforming a code base that has some problems into one that is sound and efficient. The third part is the payoff: a single chapter containing a list of heuristics and "smells" gathered while creating the case studies. The result is a knowledge base that describes the way we think when we write, read, and clean code.

Readers will come away from this book understanding

How to tell the difference between good and bad code
How to write good code and how to transform bad code into good code
How to create good names, good functions, good objects, and good classes
How to format code for maximum readability
How to implement complete error handling without obscuring code logic
How to unit test and practice test-driven development
This book is a must for any developer, software engineer, project manager, team lead, or systems analyst with an interest in producing better code.

About the Author

Robert C. "Uncle Bob" Martin has been a software professional since 1970 and an international software consultant since 1990. He is founder and president of Object Mentor, Inc., a team of experienced consultants who mentor their clients worldwide in the fields of C++, Java, C#, Ruby, OO, Design Patterns, UML, Agile Methodologies, and eXtreme programming.

Excerpt. © Reprinted by permission. All rights reserved.

Which door represents your code? Which door represents your team or your company? Why are we in that room? Is this just a normal code review or have we found a stream of horrible problems shortly after going live? Are we debugging in a panic, poring over code that we thought worked? Are customers leaving in droves and managers breathing down our necks? How can we make sure we wind up behind the *right* door when the going gets tough? The answer is: *craftsmanship*.

There are two parts to learning craftsmanship: knowledge and work. You must gain the knowledge of principles, patterns, practices, and heuristics that a craftsman knows, and you must also grind that knowledge into your fingers, eyes, and gut by working hard and practicing.

I can teach you the physics of riding a bicycle. Indeed, the classical mathematics is relatively straightforward. Gravity, friction, angular momentum, center of mass, and so forth, can be demonstrated with

less than a page full of equations. Given those formulae I could prove to you that bicycle riding is practical and give you all the knowledge you needed to make it work. And you'd still fall down the first time you climbed on that bike.

Coding is no different. We could write down all the "feel good" principles of clean code and then trust you to do the work (in other words, let you fall down when you get on the bike), but then what kind of teachers would that make us, and what kind of student would that make you?

No. That's not the way this book is going to work.

Learning to write clean code is hard work. It requires more than just the knowledge of principles and patterns. You must sweat over it. You must practice it yourself, and watch yourself fail. You must watch others practice it and fail. You must see them stumble and retrace their steps. You must see them agonize over decisions and see the price they pay for making those decisions the wrong way.

Be prepared to work hard while reading this book. This is not a "feel good" book that you can read on an airplane and finish before you land. This book will make you work, and work hard. What kind of work will you be doing? You'll be reading code--lots of code. And you will be challenged to think about what's right about that code and what's wrong with it. You'll be asked to follow along as we take modules apart and put them back together again. This will take time and effort; but we think it will be worth it.

We have divided this book into three parts. The first several chapters describe the principles, patterns, and practices of writing clean code. There is quite a bit of code in these chapters, and they will be challenging to read. They'll prepare you for the second section to come. If you put the book down after reading the first section, good luck to you!

The second part of the book is the harder work. It consists of several case studies of ever-increasing complexity. Each case study is an exercise in cleaning up some code--of transforming code that has some problems into code that has fewer problems. The detail in this section is intense. You will have to flip back and forth between the narrative and the code listings. You will have to analyze and understand the code we are working with and walk through our reasoning for making each change we make. Set aside some time because *this should take you days*.

The third part of this book is the payoff. It is a single chapter containing a list of heuristics and smells gathered while creating the case studies. As we walked through and cleaned up the code in the case studies, we documented every reason for our actions as a heuristic or smell. We tried to understand our own reactions to the code we were reading and changing, and worked hard to capture why we felt what we felt and did what we did. The result is a knowledge base that describes the way we think when we write, read, and clean code.

This knowledge base is of limited value if you don't do the work of carefully reading through the case studies in the second part of this book. In those case studies we have carefully annotated each change we made with forward references to the heuristics. These forward references appear in square brackets like this: H22. This lets you see the context in which those heuristics were applied and written! It is not the heuristics themselves that are so valuable, it is the *relationship between those heuristics and the discrete decisions we made while cleaning up the code in the case studies*.

To further help you with those relationships, we have placed a cross-reference at the end of the book that shows the page number for every forward reference. You can use it to look up each place where a certain heuristic was applied.

If you read the first and third sections and skip over the case studies, then you will have read yet another "feel good" book about writing good software. But if you take the time to work through the case studies, following every tiny step, every minute decision--if you put yourself in our place, and force yourself to think along the same paths that we thought, then you will gain a much richer understanding of those principles, patterns, practices, and heuristics. They won't be "feel good" knowledge any more. They'll have been ground into your gut, fingers, and heart. They'll have become part of you in the same way that a bicycle becomes an extension of your will when you have mastered how to ride it.

Users Review

From reader reviews:

Jennifer Carter:

Have you spare time for a day? What do you do when you have far more or little spare time? Sure, you can choose the suitable activity to get spend your time. Any person spent all their spare time to take a walk, shopping, or went to the particular Mall. How about open or read a book titled Clean Code: A Handbook of Agile Software Craftsmanship? Maybe it is to become best activity for you. You realize beside you can spend your time with your favorite's book, you can more intelligent than before. Do you agree with the opinion or you have different opinion?

Thelma Burke:

The book Clean Code: A Handbook of Agile Software Craftsmanship make one feel enjoy for your spare time. You can utilize to make your capable far more increase. Book can to be your best friend when you getting tension or having big problem with the subject. If you can make looking at a book Clean Code: A Handbook of Agile Software Craftsmanship being your habit, you can get more advantages, like add your own capable, increase your knowledge about many or all subjects. You are able to know everything if you like open and read a book Clean Code: A Handbook of Agile Software Craftsmanship. Kinds of book are a lot of. It means that, science e-book or encyclopedia or other folks. So , how do you think about this reserve?

Kenneth Vargas:

Book is to be different for each and every grade. Book for children till adult are different content. We all know that that book is very important for people. The book Clean Code: A Handbook of Agile Software Craftsmanship seemed to be making you to know about other knowledge and of course you can take more information. It doesn't matter what advantages for you. The e-book Clean Code: A Handbook of Agile Software Craftsmanship is not only giving you far more new information but also being your friend when you feel bored. You can spend your own spend time to read your e-book. Try to make relationship using the book Clean Code: A Handbook of Agile Software Craftsmanship. You never sense lose out for everything should you read some books.

Benjamin Deloatch:

This Clean Code: A Handbook of Agile Software Craftsmanship usually are reliable for you who want to be

described as a successful person, why. The reason why of this Clean Code: A Handbook of Agile Software Craftsmanship can be one of many great books you must have will be giving you more than just simple reading food but feed an individual with information that maybe will shock your preceding knowledge. This book is actually handy, you can bring it everywhere and whenever your conditions in the e-book and printed ones. Beside that this Clean Code: A Handbook of Agile Software Craftsmanship forcing you to have an enormous of experience for example rich vocabulary, giving you demo of critical thinking that could it useful in your day action. So , let's have it and revel in reading.

Download and Read Online Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin #K9TY1LWO52A

Read Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin for online ebook

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin books to read online.

Online Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin ebook PDF download

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin Doc

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin MobiPocket

Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin EPub

K9TY1LWO52A: Clean Code: A Handbook of Agile Software Craftsmanship By Robert C. Martin