



Robert C. Martin Series

Clean Craftsmanship

Disciplines, Standards, and Ethics

Foreword by **Stacia Heimgartner Viscardi**, CST & Agile Mentor

Robert C. Martin

Praise for Clean Craftsmanship

“Bob’s *Clean Craftsmanship* has done a great job explaining the purposes of agile technical practices, along with a deep historical basis for how they came into existence, as well as positioning for why they will always be important. His involvement in history and formation of agility, thorough understanding of practices, and their purposes reflect vividly throughout the manuscript.”

—Tim Ottinger, *well-known Agile Coach and author*

“Bob’s writing style is excellent. It is easy to read and the concepts are explained in perfect detail for even a new programmer to follow. Bob even has some funny moments, which pleasantly snap you out of focus. The true value of the book is really in the cry for change, for something better . . . the cry for programmers to be professional . . . the realization that software is everywhere. Additionally, I believe there is a lot of value in all the history Bob provides. I enjoy that he doesn’t waste time laying blame for how we got to where we are now. Bob calls people to action, asking them to take responsibility by increasing their standards and level of professionalism, even if that means pushing back sometimes.”

—Heather Kanser

“As software developers, we have to continually solve important problems for our employers, customers, colleagues, and future selves. Getting the app to work, though difficult, is not enough, it does not make you a craftsman. With an app working, you have passed the *app-titude* test. You may have the aptitude to be a craftsman, but there is more to master. In these pages, Bob expresses clearly the techniques and responsibilities to go beyond the *app-titude* test and shows the way of the serious software craftsman.”

—James Grenning, *author of Test-Driven Development for Embedded C and Agile Manifesto co-author*

“Bob’s one of the very few famous developers with whom I’d like to work on a tech project. It’s not because he’s a good developer, famous, or a good communicator; it’s because Bob helps me be a better developer and a team member. He has spotted every major development trend, years ahead of others, and has been able to explain its importance, which encouraged me to learn. Back when I started—apart from being honest and a good person—the idea of craftsmanship and ethics was completely missing from this field. Now, it seems to be the most important thing professional developers can learn, even ahead of coding itself. I’m happy to see Bob leading the way again. I can’t wait to hear his perspective and incorporate it into my own practice.”

—Daniel Markham, *Principal, Bedford Technology Group, Inc.*

Clean Craftsmanship: Disciplines, Standards, and Ethics

Table of Contents

Cover

Half Title

Title Page

Copyright Page

Dedication

Table of Contents

Foreword

Preface

Acknowledgments

About the Author

Chapter 1 Craftsmanship

PART I: The Disciplines

Chapter 2 Test-Driven Development

Overview

Software

The Three Laws of TDD

The Fourth Law

The Basics

Simple Examples

Stack

Prime Factors

The Bowling Game

Table of Contents

Conclusion

Chapter 3 Advanced TDD

Sort 1

Sort 2

Getting Stuck

Arrange, Act, Assert

Enter BDD

Finite State Machines

BDD Again

Test Doubles

Dummy

Stub

Spy

Mock

Fake

The TDD Uncertainty Principle

London versus Chicago

The Certainty Problem

London

Chicago

Synthesis

Architecture

Conclusion

Chapter 4 Test Design

Testing Databases

Testing GUIs

GUI Input

Test Patterns

Test-Specific Subclass

Self-Shunt

Humble Object

Test Design

Table of Contents

The Fragile Test Problem
The One-to-One Correspondence
Breaking the Correspondence
The Video Store
Specificity versus Generality

Transformation Priority Premise

{ } Nil
Nil Constant
Unconditional Selection
Value List
Statement Recursion
Selection Iteration
Value Mutated Value
Example: Fibonacci
The Transformation Priority Premise

Conclusion

Chapter 5 Refactoring

What Is Refactoring?

The Basic Toolkit

Rename
Extract Method
Extract Variable
Extract Field
Rubiks Cube

The Disciplines

Tests
Quick Tests
Break Deep One-to-One Correspondences
Refactor Continuously
Refactor Mercilessly
Keep the Tests Passing!
Leave Yourself an Out

Conclusion

Table of Contents

Chapter 6 Simple Design

- YAGNI

- Covered by Tests

 - Coverage

 - An Asymptotic Goal

 - Design?

 - But Theres More

- Maximize Expression

 - The Underlying Abstraction

 - Tests: The Other Half of the Problem

- Minimize Duplication

 - Accidental Duplication

- Minimize Size

 - Simple Design

Chapter 7 Collaborative Programming

Chapter 8 Acceptance Tests

- The Discipline

- The Continuous Build

Extreme Programming

- The Circle of Life

Test-Driven Development

Refactoring

Simple Design

Collaborative Programming

Acceptance Tests

PART II: The Standards

Your New CTO

Chapter 9 Productivity

- We Will Never Ship S**T

Table of Contents

Inexpensive Adaptability

We Will Always Be Ready

Stable Productivity

Chapter 10 Quality

Continuous Improvement

Fearless Competence

Extreme Quality

We Will Not Dump on QA

The QA Disease

QA Will Find Nothing

Test Automation

Automated Testing and User Interfaces

Testing the User Interface

Chapter 11 Courage

We Cover for Each Other

Honest Estimates

You Must Say NO

Continuous Aggressive Learning

Mentoring

PART III: The Ethics

The First Programmer

Seventy-Five Years

Nerds and Saviors

Role Models and Villains

We Rule the World

Catastrophes

The Oath

Chapter 12 Harm

Table of Contents

First, Do No Harm

- No Harm to Society
- Harm to Function
- No Harm to Structure
- Soft
- Tests

Best Work

- Making It Right
- What Is Good Structure?
- Eisenhowers Matrix
- Programmers Are Stakeholders
- Your Best

Repeatable Proof

- Dijkstra
- Proving Correctness
- Structured Programming
- Functional Decomposition
- Test-Driven Development

Chapter 13 Integrity

Small Cycles

- The History of Source Code Control
- Git
- Short Cycles
- Continuous Integration
- Branches versus Toggles
- Continuous Deployment
- Continuous Build

Relentless Improvement

- Test Coverage
- Mutation Testing
- Semantic Stability
- Cleaning
- Creations

Table of Contents

Maintain High Productivity

Viscosity

Managing Distractions

Time Management

Chapter 14 Teamwork

Work as a Team

Open/Virtual Office

Estimate Honestly and Fairly

Lies

Honesty, Accuracy, Precision

Story 1: Vectors

Story 2: pCCU

The Lesson

Accuracy

Precision

Aggregation

Honesty

Respect

Never Stop Learning

Index