

Robert C. Martin Series

# Clean Architecture

A Craftsman's Guide to  
Software Structure and Design

**Robert C. Martin**

*With contributions by James Grenning and Simon Brown*

*Foreword by Kevlin Henney*

*Afterword by Jason Gorman*



# **Clean Architecture**

# **Clean Architecture: A Craftsman's Guide to Software Structure and Design**

## **Table of Contents**

Cover

Half Title

Title Page

Copyright Page

Dedication

Contents

Foreword

Preface

Acknowledgments

About the Author

**PART I: Introduction**

Chapter 1 What Is Design and Architecture?

The Goal?

Case Study

Conclusion

Chapter 2 A Tale of Two Values

Behavior

Architecture

The Greater Value

Eisenhowers Matrix

Fight for the Architecture

# **Table of Contents**

## **PART II: Starting with the Bricks: Programming Paradigms**

### **Chapter 3 Paradigm Overview**

- Structured Programming
- Object-Oriented Programming
- Functional Programming
- Food for Thought
- Conclusion

### **Chapter 4 Structured Programming**

- Proof
- A Harmful Proclamation
- Functional Decomposition
- No Formal Proofs
- Science to the Rescue
- Tests
- Conclusion

### **Chapter 5 Object-Oriented Programming**

- Encapsulation?
- Inheritance?
- Polymorphism?
- Conclusion

### **Chapter 6 Functional Programming**

- Squares of Integers
- Immutability and Architecture
- Segregation of Mutability
- Event Sourcing
- Conclusion

## **PART III: Design Principles**

### **Chapter 7 SRP: The Single Responsibility Principle**

# **Table of Contents**

Symptom 1: Accidental Duplication

Symptom 2: Merges

Solutions

Conclusion

## **Chapter 8 OCP: The Open-Closed Principle**

A Thought Experiment

Directional Control

Information Hiding

Conclusion

## **Chapter 9 LSP: The Liskov Substitution Principle**

Guiding the Use of Inheritance

The Square/Rectangle Problem

LSP and Architecture

Example LSP Violation

Conclusion

## **Chapter 10 ISP: The Interface Segregation Principle**

ISP and Language

ISP and Architecture

Conclusion

## **Chapter 11 DIP: The Dependency Inversion Principle**

Stable Abstractions

Factories

Concrete Components

Conclusion

## **PART IV: Component Principles**

### **Chapter 12 Components**

A Brief History of Components

Relocatability

Linkers

# **Table of Contents**

Conclusion

## **Chapter 13 Component Cohesion**

The Reuse/Release Equivalence Principle

The Common Closure Principle

The Common Reuse Principle

The Tension Diagram for Component Cohesion

Conclusion

## **Chapter 14 Component Coupling**

The Acyclic Dependencies Principle

Top-Down Design

The Stable Dependencies Principle

The Stable Abstractions Principle

Conclusion

## **PART V: Architecture**

### **Chapter 15 What Is Architecture?**

Development

Deployment

Operation

Maintenance

Keeping Options Open

Device Independence

Junk Mail

Physical Addressing

Conclusion

### **Chapter 16 Independence**

Use Cases

Operation

Development

Deployment

# Table of Contents

Leaving Options Open

Decoupling Layers

Decoupling Use Cases

Decoupling Mode

Independent Develop-ability

Independent Deployability

Duplication

Decoupling Modes (Again)

Conclusion

## Chapter 17 Boundaries: Drawing Lines

A Couple of Sad Stories

FitNesse

Which Lines Do You Draw, and When Do You Draw Them?

What About Input and Output?

Plugin Architecture

The Plugin Argument

Conclusion

## Chapter 18 Boundary Anatomy

Boundary Crossing

The Dreaded Monolith

Deployment Components

Threads

Local Processes

Services

Conclusion

## Chapter 19 Policy and Level

Level

Conclusion

## Chapter 20 Business Rules

# Table of Contents

Entities

Use Cases

Request and Response Models

Conclusion

## Chapter 21 Screaming Architecture

The Theme of an Architecture

The Purpose of an Architecture

But What About the Web?

Frameworks Are Tools, Not Ways of Life

Testable Architectures

Conclusion

## Chapter 22 The Clean Architecture

The Dependency Rule

A Typical Scenario

Conclusion

## Chapter 23 Presenters and Humble Objects

The Humble Object Pattern

Presenters and Views

Testing and Architecture

Database Gateways

Data Mappers

Service Listeners

Conclusion

## Chapter 24 Partial Boundaries

Skip the Last Step

One-Dimensional Boundaries

Facades

Conclusion

## Chapter 25 Layers and Boundaries



# **Table of Contents**

- Hunt the Wumpus
- Clean Architecture?
- Crossing the Streams
- Splitting the Streams
- Conclusion

## **Chapter 26 The Main Component**

- The Ultimate Detail
- Conclusion

## **Chapter 27 Services: Great and Small**

- Service Architecture?
- Service Benefits?
- The Kitty Problem
- Objects to the Rescue
- Component-Based Services
- Cross-Cutting Concerns
- Conclusion

## **Chapter 28 The Test Boundary**

- Tests as System Components
- Design for Testability
- The Testing API
- Conclusion

## **Chapter 29 Clean Embedded Architecture**

- App-titude Test
- The Target-Hardware Bottleneck
- Conclusion

## **PART VI: Details**

### **Chapter 30 The Database Is a Detail**

- Relational Databases
- Why Are Database Systems So Prevalent?

# Table of Contents

What If There Were No Disk?

Details

But What about Performance?

Anecdote

Conclusion

## Chapter 31 The Web Is a Detail

The Endless Pendulum

The Upshot

Conclusion

## Chapter 32 Frameworks Are Details

Framework Authors

Asymmetric Marriage

The Risks

The Solution

I Now Pronounce You

Conclusion

## Chapter 33 Case Study: Video Sales

The Product

Use Case Analysis

Component Architecture

Dependency Management

Conclusion

## Chapter 34 The Missing Chapter

Package by Layer

Package by Feature

Ports and Adapters

Package by Component

The Devil Is in the Implementation Details

Organization versus Encapsulation

# **Table of Contents**

Other Decoupling Modes

Conclusion: The Missing Advice

**PART VII: Appendix**

**Appendix A: Architecture Archaeology**

**Index**