

# **The Software Craftsperson's Pocket Book**

A Mind-map Based Guide  
to Software Excellence

**Srihari Sridharan**





## Preface

After completing the second edition of "Craft Your Code," I found myself exploring ideas for my next book. While several concepts crossed my mind, I felt compelled to maintain continuity with the themes explored in my previous book. Hence, the book you hold in your hands at this moment came into existence. Even if you haven't had the opportunity to read my earlier book, don't worry, this book serves as both a roadmap for budding engineers and a refresher for seasoned professionals.

One of the main motivations behind writing this book originates from my experience, belonging to a generation of developers who witnessed the evolution of programming languages, frameworks, and IDEs firsthand. Reflecting on the nascent stages of my career, C# was in its infancy at version 1.1, while Java was at version 2. JavaScript, was used in web pages (not on the server). Now JavaScript is widespread alongside

TypeScript and other languages. Notably, the robust IDEs of today, did not have functionalities we now take for granted, such as refactoring methods, renaming variables and classes. I was using Visual Studio 2003. Features as fundamental as generics debuted only with the advent of C# 2.0. Today, as I pen this preface, C# has matured to version 12, while Java boasts Version 21.

Welcome to the "**The Software Craftsperson's Pocket Book**". In software development, craftsmanship serves as the foundation of excellence. This pocket book endeavors to capture the essence of software craftsmanship into a succinct and accessible format, catering to novices and seasoned practitioners alike. Beyond mere coding proficiency, craftsmanship embodies a holistic approach to software development, encapsulating principles, practices, and a mindset geared towards perpetual refinement. "Craft Your Code" taught you how to write high quality code, whereas this book takes you to the next level. In this book, you will find a collection of resources, insights, strategies, and techniques to refine your craft and elevate your skills to the next level.

Whether you are a software engineer, a team leader, or an aspiring craftsperson, this book

serves as your companion on the journey towards mastery. It contains structured chapters, followed by a mind map, covers an array of topics encompassing code quality, design principles, and effective collaboration and communication strategies. As you start this journey on software craftsmanship, please bear in mind that mastery is not a destination but a continuum of learning and growth. May this pocket book serve as your guide, empowering you to unlock your fullest potential as a software craftsperson.

Happy crafting!

**Author's Note:** Considering the size of this book, some diagrams and mindmaps are printed in landscape for improved readability.

# Table of Contents

<b>Preface .....</b>	<b>i</b>
<b>Table of Contents .....</b>	<b>iv</b>
<b>About the Author .....</b>	<b>viii</b>
<b>The idea behind the name .....</b>	<b>x</b>
<b>1. Software Development &amp; Software</b>	
<b>Craftsmanship .....</b>	<b>1</b>
<b>Development vs. Craftsmanship.....</b>	<b>1</b>
<b>Getting your code to work is just like setting a very low bar .....</b>	<b>2</b>
<b>Software is ever changing and never finished .</b>	<b>3</b>
<b>You are developing the legacy systems of the future .....</b>	<b>4</b>
<b>The cycle continues.....</b>	<b>5</b>
<b>Bringing It All Together .....</b>	<b>6</b>
<b>Further Reading and References.....</b>	<b>8</b>
<b>Contributing to Open Source and Communities .....</b>	<b>9</b>
<b>Contributing to open-source.....</b>	<b>10</b>
<b>Understanding the Importance .....</b>	<b>10</b>

Finding Opportunities .....	11
Types of Contributions .....	11
Best Practices for Contribution .....	11
Growing Your Network .....	12
Giving Back .....	12
<b>Building communities and sharing knowledge</b> .....	<b>14</b>
Importance of Learning Communities .....	15
Building and Nurturing Communities.....	16
Sharing Knowledge.....	17
Embracing Diversity and Inclusion .....	17
Leveraging Online Platforms .....	18
Promoting Lifelong Learning.....	18
<b>Bringing It All Together .....</b>	<b>19</b>
<b>Further Reading and References.....</b>	<b>21</b>
<b>3. The Software Craftsperson's Toolbox.....</b>	<b>23</b>
<b>Test-Driven Development.....</b>	<b>24</b>
<b>Behavior-Driven Development.....</b>	<b>27</b>
<b>Domain-Driven Design.....</b>	<b>34</b>
Ubiquitous Language .....	35
Bounded Contexts.....	35
Strategic Design .....	37
Tactical Design .....	37
<b>Pair Programming and Mob Programming ....</b>	<b>37</b>
<b>Programming Paradigms, Patterns and Principles .....</b>	<b>41</b>



SOLID Principles.....	42
Package Cohesion Principles.....	43
Package Coupling Principles .....	43
<b>Legacy Remediation .....</b>	<b>44</b>
<b>Craft Your Code - Writing Clean Code .....</b>	<b>47</b>
<b>Bringing It All Together .....</b>	<b>48</b>
<b>Further Reading and References.....</b>	<b>50</b>
<b>4. Test-Driven Development .....</b>	<b>51</b>
<b>What Is TDD? .....</b>	<b>51</b>
Things to keep in mind .....	55
Advantages .....	55
Disadvantages .....	56
<b>Red - Green - Refactor .....</b>	<b>57</b>
<b>Writing Effective Tests.....</b>	<b>60</b>
<b>Approaches to TDD .....</b>	<b>61</b>
Inside-Out Approach .....	62
Outside-In Approach .....	63
<b>Exercise .....</b>	<b>64</b>
<b>Bringing It All Together .....</b>	<b>64</b>
<b>Further Reading and References.....</b>	<b>66</b>
<b>5. Behavior Driven Development.....</b>	<b>67</b>
<b>What is Behavior Driven Development? .....</b>	<b>67</b>
<b>Benefits of Behavior Driven Development.....</b>	<b>71</b>
<b>Disadvantages of Behavior Driven Development .....</b>	<b>73</b>

<b>An example of BDD Specification .....</b>	<b>76</b>
<b>Exercise .....</b>	<b>76</b>
<b>Bring It All Together .....</b>	<b>76</b>
<b>Further Reading and References.....</b>	<b>78</b>
<b>6. Mutation Testing .....</b>	<b>79</b>
<b>What is Mutation Testing?.....</b>	<b>79</b>
Key Concepts .....	80
Mutants.....	81
Killed .....	81
Survived.....	81
Mutation Score .....	81
Timeout .....	81
No Coverage .....	82
Runtime Errors.....	82
Compile Errors.....	82
Detected and Undetected Mutants .....	82
<b>How Stuff Works.....</b>	<b>82</b>
Introducing Stryker .....	83
<b>Bringing It All Together .....</b>	<b>92</b>

## About the Author

Srihari Sridharan is a software architect, coach, mentor and leader with a hands-on approach towards technology and leadership. He is a renowned speaker, conducting courses and delivering talks on leadership, transforming teams, software craftsmanship and writing clean code. Srihari's areas of expertise encompass mentoring, coaching, leadership, software architecture, design and development, building and transforming teams. He is a firm believer in leading by example, Agile methodologies, Extreme Programming (XP), Test-Driven Development (TDD), and the principles of writing clean code and refactoring.

Apart from his role in a leading company that provides premium software services to global clients, Srihari is a reviewer and a senior technical proof-reader for Manning Publications Co. He was the technical reviewer for more than the 15 Manning titles and proofread more than 5 of them. He actively serves as a member of the 'Board of Studies - Department of Information Technology' at B.S Abdur Rahman Crescent

Institute of Science & Technology, where he strives to bridge the gap between academic curriculum and industry expectations. His responsibilities include framing and reviewing syllabi for various subjects.

## The idea behind the name

The deliberate choice of the term "craftsperson" over "craftsman" underscores my commitment to inclusivity and gender neutrality throughout the title, content, and the publication. It is my belief that inclusivity and diversity are not mere buzzwords but integral tenets to be upheld in our shared pursuit of software as a craft. I feel people should be respected, paid, and treated well regardless of gender and background. I am confident readers will find valuable insights within these pages to enrich their software development endeavors.

Let us get started!



