# Agile Principles, Patterns, and Practices in C#:

This book is an updated C# version of the well-known Book called **Agile Software Development: Principles, Patterns, and Practices**, Robert C. Martin helped bring Agile principles to tens of thousands of Java and C++ programmers.

This book presents a series of case studies illustrating the fundamentals of Agile development and Agile design, and moves quickly from UML models to real C# code. The introductory chapters lay out the basics of the agile movement, while the later chapters show proven techniques in action. The book includes many source code examples.

## After reading this book, we will gain the below:

* Agile principles, and the fourteen practices of Extreme Programming
* Spiking, splitting, velocity, and planning iterations and releases
* Test-driven development, test-first design, and acceptance testing
* Refactoring with unit testing
* Pair programming
* Agile design and design smells
* The five types of UML diagrams and how to use them effectively
* Object-oriented package design and design patterns
* How to put all of it together for a real-world project

## Forward:

work with individuals obsessed with shipping software, develop with lightweight processes that are tuned to each team, and adapt constantly.

Double-clicking on developers from our teams reveals individuals who consider programming the focus of development. Not only do they write code; they digest it constantly to maintain an understanding of the system. Validating designs with code provides feedback that's crucial for getting confidence in a design. At the same time, our developers understand the importance of patterns, refactoring, testing, incremental delivery, frequent builds, and other best-practices of XP that have altered the way we view methodologies today.

Skill in this style of development is a prerequisite for success in projects with high technical risk and changing requirements. Agile development is low-key on ceremony and project documentation, but it's intense when it comes to the day-to-day development practices that count. Putting these practices to work is the focus of this book.

## Preface:

Martin Saied:

Agile development is the ability to develop software quickly, in the face of rapidly changing requirements. To achieve this agility, we need to use practices that provide the necessary discipline and feedback. We need to employ design principles that keep our software flexible and maintainable, and we need to know the design patterns that have been shown to balance those principles for specific problems. This book is an attempt to knit all three of these concepts together into a functioning whole.

Micah Saied:

In my classes, .NET programmers tend to be weaker in agile software practices, design patterns, design principles, and so on. Often in my classes, the .NET programmers had never heard of these fundamental concepts. This must change.

The first edition of this book, Agile Software Development: Principles, Patterns, and Practices, by Robert C. Martin, my father, was published in late 2002 and won the 2003 Jolt Award. It is an enjoyable book, celebrated by many developers. Unfortunately, it had minor impact on the .NET community. Even though the content of the book is equally relevant to .NET, few .NET programmers have read it.

It is my hope that this .NET edition acts as a bridge between .NET and the rest of the developer community.

## About This Book:

In the early 1990s Bob wrote Designing Object-Oriented C++ Applications Using the Booch Method. The book you are reading started out as a second edition to Designing, but that's not how it turned out. Very little remains of the original book in these pages due to the experience and the new technologies.

This book has a repeating pattern: a series of case studies of varying sizes. Some are very small, and some require several chapters to describe. Each case study is preceded by material that is meant to prepare you for it by describing the object-oriented design principles and patterns used in that case study.

This book is organized into four sections and two appendixes:

1. Agile Development:

Agile Development, describes the concept of agile development. It starts with the Manifesto of the Agile Alliance, provides an overview of Extreme Programming (XP), and then goes to many small case studies that illuminate some of the individual XP practices, especially those that have an impact on the way we design and write code.

1. Agile Design:

Agile Design, talks about object-oriented software design: what it is, the problem of and techniques for managing complexity, and the principles of object-oriented class design. The section concludes with several chapters that describe a pragmatic subset of UML.

1. The Payroll Case Study:

The Payroll Case Study, describes the object-oriented design and C++ implementation of a simple batch payroll system. The first few chapters in this section describe the design patterns that

the case study encounters. The final chapter is the full case study, the largest and most complete one in the book.

1. Packaging the Payroll System:

Packaging the Payroll System, begins by describing the principles of object-oriented package design and then goes on to illustrate those principles by incrementally packaging the classes from the previous section. The section concludes with chapters that describe the database and UI design of the Payroll application.

1. Appendix A.
2. Appendix B.

# Section I: Agile Development:

## Intro:

Process and technology are a second-order effect on the outcome of a project. The first-order effect is the people.

Build a powerful team that apply a process is better than applying it with a bad team.

## Chapter 1. Agile Practices:

# The Clean Architecture:

