to be agile
lessen the learning curve

Scrum Software Developer Essentials

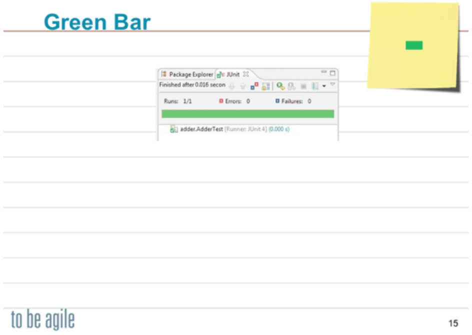
This immersive, twenty-four-hour, hands-on software developer training course presents key developer practices from Scrum and Extreme Programming (XP) that will enable you to build higher-quality software more rapidly and with fewer defects and make you a more valuable contributor on any development team. You'll learn essential practices and expert techniques that support the creation of more maintainable, extensible software. You'll see how test-first development informs design decisions and gain the tools for exploring the best design tradeoffs for the tasks at hand. You'll discover the secrets to finding patterns in problems, find easy ways to identify abstractions, and master several techniques for emerging designs.

The course includes six hands-on programming labs in which you'll get to demonstrate the techniques you've learned by building a system and seeing your design unfold. Examples apply to any programming language and the exercises can be done using Java or C Sharp. By the end of class, you will have a thorough understanding of how to use Scrum to improve your software development, along with the knowledge and skills to make it happen.

Course Benefits

Completing this course will give you a deeper understanding of how to build higher-quality software on a Scrum development team and enable you to:

- Employ expert techniques for analysis and design
- Read and write the most useful UML diagrams
- Estimate development tasks more accurately
- Deliver valuable software in sprints
- Create flexible designs that can be easily changed
- Apply simple strategies for refactoring legacy code
- Identify code qualities that improve maintainability
- Recognize pathologies in code and how to fix them
- Experience emerging designs using test-first development


Green Bar

```
Finished after 0.016 secon
Runs: 1/1    Errors: 0    Failures: 0
adder.AdderTest [Runner: JUnit 4] (0.000 s)
```

Who Should Take This Course

This training is for all technical team members and has the greatest impact when the entire technical team can attend. This course will benefit architects; designers; developers; development managers; product managers; software engineers/programmers; testers; technical analysts; technical leads; and QA engineers. Familiarity with basic Object-Oriented (OO) concepts and terminology is recommended. Those who participate in the programming exercises and fulfill the additional training requirements are eligible for becoming Certified Scrum Developers and require the ability to write simple programs in Java or C Sharp.

Agenda

In Three Full-Day (8-Hour) In-Person Sessions

Session 1 — Analysis

Introduction: Scrum versus Waterfall development

Development Essentials: Scrum and XP practices

Test-Driven Development: TDD driving design

Lab 1: “Iteration Zero—Intro and set up of case study

Lab 2: “Coding Test First—Experiencing TDD

Session 2 — Design

Design Review: Group discussion of previous day’s labs

CREATE Code Qualities: Critical code qualities; recognizing and getting more out of them

Development Practices: Simple techniques of rapidly building quality software

Lab 3: “Adding Optional Behaviors—Using requirements to find patterns in problems

Lab 4: “Encapsulating Complex Business Rules—Using compound patterns

Session 3 — Development

Design Review: Group discussion of the patterns in the previous day’s labs

Testing Techniques: Advanced techniques for writing more-testable software

Refactoring and Emerging Systems: Techniques for refactoring legacy code

Lab 5: “Mocks and Shunts—Techniques for writing mocks and injecting dependencies

Lab 6: “Refactoring Legacy Code—How to clean up code by identifying code smells

Concluding Remarks

In Five Half-Day (4-Hour) Online Sessions Session

Session 1A — Analysis

Introduction: Scrum versus Waterfall development

Development Essentials: Scrum and XP practices

Test-Driven Development I: TDD driving design

Test-Driven Development II: TDD walkthrough

Session 1B — Lab 1 & 2

Lab 1: “Iteration Zero—Intro and set up of case study

Lab 2: “Coding Test First—Experiencing TDD

Session 2A — Design

Design Review: Group discussion of previous day’s labs

CREATE Code Qualities: Critical code qualities; recognizing and getting more out of them

Development Practices: Simple techniques of rapidly building quality software

Session 2B — Lab 3 & 4

Lab 3: “Adding Optional Behaviors—Using requirements to find patterns in problems

Lab 4: “Encapsulating Complex Business Rules—Using compound patterns

Session 3A — Development

Design Review: Group discussion of the patterns in the previous day’s labs

Testing Techniques: Advanced techniques for writing more-testable software

Refactoring and Emerging Systems: Techniques for refactoring legacy code

Optional Session 3B — Lab 5 & 6

Lab 5: “Mocks and Shunts—Techniques for writing mocks and injecting dependencies

Lab 6: “Refactoring Legacy Code—How to clean up code by identifying code smells

Concluding Remarks

Your Instructor

My continuing passion for software design and construction has led me to train more than 10,000 professional software developers for clients that have included Fortune 500 firms such as Microsoft, IBM, Yahoo, Boeing, AT&T, Sprint, Medtronic, SunGard, State Farm, Vanguard, and Weyerhaeuser. As a longtime IBM consultant, I trained software engineers around the globe, giving them the skills to write the next generation of applications and operating system software while earning one of the highest satisfaction ratings in the history of IBM education. Since 2006, I've devoted my consulting practice to providing organizations with training and coaching for software developers and teams transitioning to Agile, Scrum, and Extreme Programming practices.

Praise for David's Training

"This was a great class! First, David distills a vast amount of Agile related information into successful practices with direct application in our work. Second, the hands-on labs get us over the learning hurdles of TDD implementation as quickly as humanly possible. Lastly, David is clearly passionate in his concern with the overall condition of code quality in the software industry. He will go to no end with his class, we voted to do two fourteen hour days to get the lab knowledge and David supported us and stayed with us the whole time. Thank you!
—Mike Jassmann, IT Manager

"Not knowing what the latest tools to use or books to read, this class allowed us to understand the latest practices and tools and how to apply them in our current working environment."
—Doug Parris, Senior Software Engineer

"This is a great class that will start you on the path to becoming a great programmer and an invaluable asset to your company."
—Vincent Quiles, Programmer Analyst

Certification

This course is part of our Certified Scrum Developer Essentials training, and satisfies the three-day technical training required to become a Certified Scrum Developer through the Scrum Alliance. This course counts for 24 Professional Development Units (PDUs). See http://ToBeAgile.com/faq for more details.