

Hands-On: Extreme Programming Practices (HEPP)

In this twenty-five-hour, advanced, hands-on software developer training course, you'll learn key developer skills from Scrum and Extreme Programming (XP), enabling you to build higher-quality software more rapidly and with fewer defects. This class is delivered online in five half-day sessions.

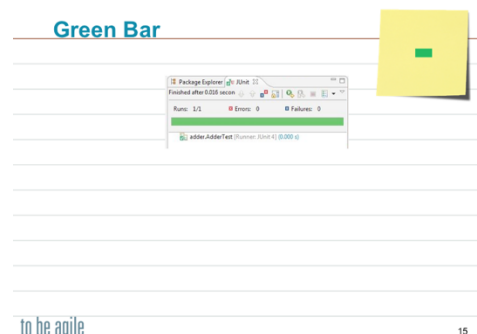
You'll master core developer practices such as BDD, TDD, refactoring, and emergent design by applying these practices to a software development project where you and your teammates will build the core of an application program. During this exercise, you'll experience how writing the test first guides you in creating testable designs and catches mistakes before you even recognize them as bugs. You'll also discover and code design patterns as part of building this application. Programming labs can be done in Java or C Sharp.

This class also includes five lectures on advanced developer practices, including writing the test first, creating code quality, developer practices, advanced testing techniques, doing emergent design, and cleaning up legacy code. By the end of this class, you will have applied the principles and practices of Extreme Programming to build a software development project that would normally take several weeks to complete in a fraction of the time.

Course Benefits

This hands-on Advanced Developer Essentials (ADE) course will give you an experience of building software incrementally and emerging designs on a development team that will enable you to:

- Master techniques for doing analysis and design
- Apply simple strategies for refactoring legacy code
- Identify code qualities that improve maintainability
- Experience emerging designs using test-first development
- Understand how to use test-first development to drive design
- Efficiently use TDD's red-green-refactor cycle
- Work effectively to refactor legacy code
- Exercise techniques to test untestable code
- Collaborate successfully with pair programming
- Avoid upfront overdesign and practice just-in-time development
- Write software that supports an iterative process without excessive rework
- Support collaborative code ownership and embrace a common aesthetic
- Refactor to patterns and emerge designs in iterative development
- Implement techniques for recognizing and managing technical debt
- Recognize how test-driven development informs design decisions



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.

This course includes twenty hours of programming labs in Java or C Sharp. Non-technical team members are welcome to attend and work alongside technical team members.

Agenda

<p>Session 1: Test-First Lecture: Test-First Development Lab 1: Setting Up</p> <p>Session 2: Techniques Lecture: CREATE Code Quality Lab 2: Writing the Test First</p> <p>Session 3: Practices Lecture: Development Practices Lab 3: Varying Behavior</p>	<p>Session 4: Testing Lecture: Testing Techniques Lab 4: Optional Behaviors</p> <p>Session 5: Refactoring Lecture: Emergent Design Lab 5: Mocks and Shunts</p>
--	--

Your Instructor; David Bernstein



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 technical 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