

## Design Pattern Developer Essentials

This engaging one-day training presents a core set of design patterns every developer should be familiar with and explores the forces underlying patterns that make them valuable for building software.

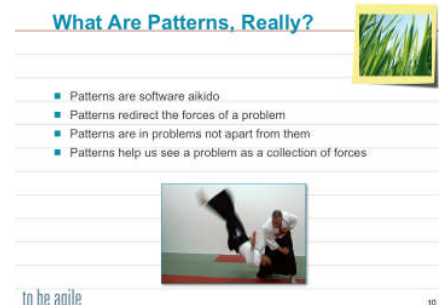
You'll see how understanding patterns reveals the essence of object-oriented thinking and creates a useful context for solving a variety of software problems. You'll come to understand patterns as more than just "reusable solutions to common problems within a given context" and recognize them as a collection of forces that help you penetrate deeper into problems, discovering elegantly simple solutions that make your software more robust and easier to maintain. The expanded view of patterns you'll gain from this course will help you encapsulate and abstract virtually any problem with maximum flexibility and without over-complicating the solution. You'll learn to make better coding choices and will master a shared vocabulary for talking about design that dramatically improves inter-team communication.

By the end of this training, you'll be armed with several new, effective tools for solving design problems that will empower you to produce immediate improvements in the quality of the software you design and build.

### Course Benefits

Completing this course will give you a deeper understanding of the object-oriented development paradigm, and enable you to:

- Explain what patterns are and know when to use them
- Read and write the 3 most-important UML diagrams
- Employ Agile principles to create higher-quality code
- Use a common vocabulary for communicating designs
- Adopt simple methods to find patterns in problems
- Understand patterns by what they encapsulate
- Apply patterns just-in-time to avoid over-design
- Appreciate the value of shared coding standards
- Contribute to design reviews and evaluate others' designs
- Gain several techniques for doing effective analysis and design
- Master techniques for emerging designs in iterative development

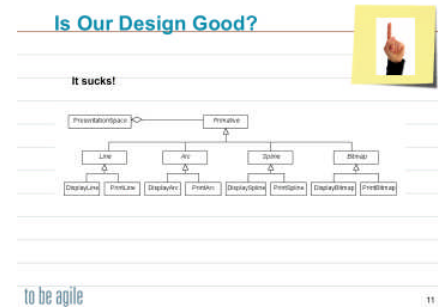


### Who Should Take This Course

This training will benefit all team members including architects; business analysts; DBAs; designers and developers; development managers; directors; documentation specialists; operations and support staffers; product and project managers; software engineers/programmers; testers and QA engineers; and technical writers, analysts and leads.

## Agenda

- **Introduction:** Purpose, objectives, and logistics.
- **Paradigms, Principles, and Perspectives:** Agile design; Gang of Four's advice; design principles.
- **Software Patterns:** Misconceptions; forces.
- **Encapsulating Variable Behavior:** Strategy, Template-Method, State, and Bridge patterns.
- **Encapsulating Foreignness:** Adapter and Façade.
- **Design Exercise Case Study:** Presentation and discussion of a representative design problem, and the use of patterns to solve it; examining the tradeoffs of different patterns.
- **Encapsulating Sequence and Cardinality:** Proxy, Decorator, Chain of Responsibility.
- **Encapsulating Construction:** The importance of factories for object construction, and how patterns teach us to separate use from creation; examination of Factory-Method, Abstract Factory, and Singleton.
- **Final Exercise:** Review of the case study introduced in the morning, and solving it by refactoring to the patterns in the problem.



## Your Instructor



David Bernstein's continuing passion for software design and construction has led him to train more than 7,500 developers in the last 23 years for clients that have included Fortune 500 firms such as Microsoft, IBM, Yahoo!, Boeing, AT&T, Sprint, Medtronic, SunGard, State Farm, MetLife and Weyerhaeuser. As a longtime IBM consultant, David 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. In the past five years, he has focused on providing organizations with training and coaching for software developers and teams transitioning to Agile and Scrum.

## Certification

This course, part of our comprehensive Certified Scrum Developer Essentials Training, satisfies the one-day elective training requirement for becoming a Certified Scrum Developer (CSD) through the Scrum Alliance. This course counts for 8 Professional Development Units (PDUs). See <http://ToBeAgile.com/faq> for more details about this new certification.

## Praise for this Training

"I learned how to think effectively about code quality. I learned principles and practices of effective design. It explains patterns in plain English. David's teaching style is very engaging and clear. I would highly recommend this class to any level of developer." – Todd Froyland, Senior Tech Lead/Architect