

*A
Practical
Guide
to*

FEATURE-DRIVEN DEVELOPMENT



- ▶ Combine the speed and flexibility of agile methods with *enterprise-class scalability*!
- ▶ Hands-on coverage of the entire project lifecycle
- ▶ Modeling, feature lists, planning, design, and software construction
- ▶ Adapt Feature-Driven Development to your own organization and projects

STEPHEN R. PALMER • JOHN M. FELSING

THE COAD SERIES

A Practical Guide to Feature-Driven Development

*Stephen R. Palmer
and John M. Felsing*

Foreword by Peter Coad



Prentice Hall PTR
Upper Saddle River, NJ 07458
www.phptr.com

Library of Congress Cataloging-in-Publication Data is available.

Production Supervision: *Donna Cullen-Dolce*
Acquisitions Editor: *Paul Petralia*
Manufacturing Buyer: *Alexis Heydt-Long*
Cover Design: *Nina Scuderi*
Cover Design Director: *Jerry Votta*
Marketing Manager: *Debby van Dijk*
Cover art: "Gravity" by *M.C. Escher*. Copyright 2001 *Cordon*
Art—*Baasm—Holland*. All rights reserved.



Copyright © 2002 by TogetherSoft Corporation
Prentice-Hall Inc.
Upper Saddle River, NJ 07458

The Coad Letter® is a registered trademark of Object International, Inc.
Together® is a registered trademark of TogetherSoft Corporation.
TogetherSoft® and Together ControlCenter™ are trademarks of TogetherSoft Corporation.
Post-It® is a trademark of 3M.
UML® is a trademark of Object Management Group.
All other products mentioned in this book are the trademarks and service marks of their respective companies, organizations, or owners.

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact:
Corporate Sales Department, Phone: 800-382-3419; Fax: 201-236-7141;
E-mail: corpsales@prenhall.com; or write: Prentice Hall PTR, Corp. Sales Dept.,
One Lake Street, Upper Saddle River, NJ 07458.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America
ISBN 0-13-067615-2

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de México, S.A. de C.V.
Pearson Education—Japan
Pearson Education Malaysia, Pte. Ltd.

A Practical Guide to Feature-Driven Development

The Coad Series

Peter Coad, *Series Editor*



- Dave Astels, Granville Miller, Miroslav Novak
A Practical Guide to eXtreme Programming
- Andy Carmichael, Dan Haywood
Better Software Faster
- Donald Kranz, Ronald J. Norman
A Practical Guide to Unified Process
- Jill Nicola, Mark Mayfield, Michael Abney
Streamlined Object Modeling: Patterns, Rules, and Implementation
- Stephen R. Palmer, John M. Felsing
A Practical Guide to Feature-Driven Development
- Jo Ellen Perry, Jeff Micke
How to Get the Most Out of the Together ControlCenter

About the Series

The Coad Series' mission statement is: Improving the ways people work together. Each book in the series delivers practical keys for building lasting value into business by building people and their skills.

Peter Coad personally selects authors and books for this series—and works on a strategic level with each author during the development of his book.

About the Series Editor

Peter Coad is a business builder, model builder, and thought leader. As business builder, Peter leads TogetherSoft Corporation (www.togethersoft.com), growing the company nearly 12 times revenue in two years. As model builder, Peter has built hundreds of models for nearly every business imaginable, ever focusing on building competitive advantages into businesses. As thought leader, Peter writes books (six to date) and speaks at events worldwide. You can contact Peter at peter.coad@togethersoft.com.

From Steve Palmer
To Suman, Mark, and Jared
For everything you give me. I don't deserve you.

From Mac Felsing
To Gina, Alex, Emilie, and Desiree
You give me determination, strength, focus, and above all, Love.



Acknowledgments

We need to thank all our friends, families, and colleagues past and present for their support, input, feedback and encouragement while writing this book.

We would especially like to thank Jeff De Luca and Peter Coad for first writing about Feature-Driven Development in their book *Java Modeling in Color with UML* [Coad99]. Also Jeff for his ability, courage, and commitment in creating a project environment where this sort of innovation could flourish, and Pete for so many insights, patterns, strategies, and techniques that made OO software analysis and design so accessible.

Thanks to the original cast of the PowerLender software development team in Singapore, especially Tan Siow Boon, Yeo Sai Cheong, Ajay Kumar Rana, Ju-Lia Tan, and James Tan, first for their willingness to adopt a new way of working, and second for their help in learning to apply FDD day after day for so many months. Too many to name individually but you know who you are.

Thanks to all the mentors at TogetherSoft who have contributed ideas, thoughts, and suggestions through countless discussions, e-mail exchanges and newsgroup threads. Special thanks to Lisa Juliani, our old "Mentor Operations Manager (MOM)," for helping get this book project started, and to Ken Ritchie, Greg Cathcart, Dan Massey, and Scott Schnier for their reviews, and continued encouragement and enthusiasm.

Also thanks to Paul Petralia, Donna Cullen-Dolce, and the rest of the team at Prentice Hall for the countless number of tasks, and their expert help and advice in turning our raw word-processor files into a real book. (As well as putting up with the quirks, foibles, and delays associated with working with two overworked, creative individuals.)

Finally, enormous thanks to our long-suffering wives for allowing us to neglect them for so many evenings and weekends to write these pages.



Foreword

To build anything of lasting value—software, systems, companies—building people must be at the heart-and-soul of all you do. “Process” is all about improving the ways people work together. A process is the family recipe, the secret sauce, the competitive edge, of human endeavor.

Everyone has his own collection of processes. So does every team. Making processes—good or bad—explicit is the first step to understanding what is going on, seeing what works and what doesn’t, and looking with eyes of wisdom as to what is going on in one of life’s many situations. Considering a process—the ways you are working together with others—is taking your thinking and going one level up. Feel pressure? Go one level up and take a look: Things begin to calm down right away; you begin to see more clearly. Feel panicked about what to do when, or what is most important? Go one level up. Having problems with a peer and just don’t know where to turn? Go one level up. Having problems producing better software with others? Go one level up.

“Process” is about moving one level up, gaining new insights, inviting others to join you one level up to make strategic moves that will make the day-to-day realities so much more effective, palatable, and dare I say it: fun, rewarding, and satisfying.

Process, going one level up, is how one begins to understand what one is doing, what others are doing, what the people issues and dynamics are, what the bottlenecks and accelerators are, what the overall context is, what the expanded context is, and what to do about it all. One level up.

Problems happening? Go one level up and find out what to do. Goodness happening? Go one level up and discover ways you can engender goodness all the more.

May we all have the courage to reach one level up, that we might better connect with others, build meaningful relationships, build people, build lasting value—and truly succeed.

That brings us to the topic of this book, the topic of Feature-Driven Development (FDD). FDD is a collection of best practices, set into the context of an overall process, one level up from the day-to-day urgency—

at times even panic—experienced by business people and developers working together on mission-critical software-development projects.

A *Practical Guide to Feature-Driven Development* reveals the process, the secret recipe, that TogetherSoft and its clients use to produce world-class software. Jeff De Luca and I first described the recipe two years ago, in a short chapter in the book *Java Modeling in Color with UML*. Steve and Mac have gone far and beyond that initial description. FDD, as described herein, is significantly advanced by these two. I am most grateful for the work they have done and the many insights they chose to share.

FDD itself is one level up.

Read the book. Consider the recipe. Try it out. Adapt it as you would with any family recipe, making it your own, making it best fit the ingredients you have on hand. Read. Enjoy. And treasure the insights from Palmer and Felsing, one level up.

Sincerely,
Peter Coad
CEO and President, TogetherSoft Corporation
peter.coad@togethersoft.com



Preface

Feature-Driven Development (FDD) is a process designed and proven to deliver frequent, tangible, working results repeatedly. This is the first book to spell out the day-to-day details of using FDD on a real project, giving development team leaders all the information they need to apply FDD successfully to their situations.

What FDD Is!

FDD is a straightforward approach to producing systems that uses simple, easy-to-understand and easy-to-implement methods; problem-solving techniques; and reporting guidelines providing every stakeholder of a project with the information they need to make sound, timely decisions.

Programmers are provided with the information and supporting infrastructure they need to produce applications. Team leaders and managers get timely information about their teams and projects that allows them to reduce the project *risk*. Project managers and executive sponsors see the current project status and trouble areas so that they can actually make timely, informed decisions in a controlled, planned manner (no knee-jerk reactions). Reporting becomes easy, relatively painless, timely, and *accurate*!

Users (customers, sponsors, end users) can actually see areas of their business automated as the project progresses and give early, constructive feedback about the system while it is being developed. At the same time, the development team has the tools and information it needs to control “scope creep!”

FDD is *not* yet another process that takes up resources, time, and money but just doesn't produce the needed results. It is not another method whereby administrators, bureaucrats, and process-centric fanatics can focus everyone's time and energy on producing reams of printouts and signatures with nothing to show for it. FDD is not another set of process volumes that will sit on your shelf, collecting dust and impressing your supervisor, co-workers, and significant others with your knowledge of another less-than-useful set of directions for producing systems.

What FDD Is Not!

Why Should I Read this Book? (What's in it for Me?)

If any of the following questions apply to you, you will find the answers you are looking for on the following pages:

- Are you in charge of a group assigned to deliver a critical system with limited resources and short deadlines, and need help to organize the project?
- Are you tired of filling out so many process forms and reviewing so many documents that you don't have time to do your real work?
- Are you frustrated at the unfulfilled promises of process initiatives that claim to provide a way to deliver quality results repeatedly?
- Are you looking for a more efficient way to organize your team to streamline productivity, cut down on unnecessary interruptions, and improve the accuracy of reporting for your projects?
- Are you currently working on a project that is in trouble because the team has failed to produce a system?
- Do you want to add to the tools and techniques in your project management or team leader toolbox?

Prego (It's in There!)

Although FDD was first introduced in print in *Java Modeling in Color with UML* [Coad 99], we give a more thorough coverage of the topic. We point out the critical tips for success, as well as the pitfalls that may not be apparent from a cursory glance. Contents include:

- Who should use FDD
- The roles, artifacts, goals, and timelines
- Why FDD includes the practices and techniques that it does
- The driving forces behind the development of FDD
- Adapting the use of FDD to different styles of projects

FDD blends a number of industry-recognized best practices used successfully by Peter Coad, Jeff De Luca, and others in their consultancies. These practices are all driven from a client-valued feature perspective. It is the potent combination of these techniques and practices that makes FDD so compelling.

John M. Felsing (mfelsi01@sprintspectrum.com)
Stephen R. Palmer (sp@togethersoft.com)
December 2001



Introduction

Answers: Why this book?

*For enterprise-component modeling to be successful,
it must live and breathe within a larger context, a
software development process.*

Coad, LeFebvre, De Luca [Coad 99]

Published in 1999, *Java Modeling in Color with UML* [Coad 99], often referred to as the “coloring book,” devotes the first five of its six chapters to a discussion of a truly significant new object modeling technique called *modeling in color*. Those first five chapters are almost, but not quite, totally irrelevant to the discussion in the rest of this book. More relevant here—much more relevant—are the contents of that sixth chapter, hidden away at the back of the book. Chapter 6 of the coloring book was the first attempt to describe in print a particularly successful way of building complex software systems by significantly large development teams with members of varying talent, experience, background, and culture. The software development process that it described is called *Feature-Driven Development* (FDD), for want of a better name.

What has become known as FDD was derived from work done on a large software development project in Singapore where Jeff De Luca was Project Manager, Peter Coad was Chief Architect, and Stephen Palmer (author) was Development Manager. The project was very ambitious, with a highly complex problem domain spanning three lines of business, from front office automation to backend legacy system integration.

The bank had already made one attempt at the project and failed. The project had inherited a skeptical user community, supportive but wary upper management, and what was left of a demoralized development team.

One of the key areas of risk that Jeff identified was the complexity of the problem domain. This demanded a level of ability and experience in domain object modeling beyond that of the current team. To mitigate

A Project in Trouble

this risk, Jeff persuaded Peter Coad to come to Singapore and work with the team to produce a resilient and flexible domain object model for the project.

To make a long story short, the result of Jeff, Peter, and others working together on this project was the discovery of the modeling in color technique and the creation of the FDD process—an agile, adaptive software development process that:

- Is highly iterative
- Emphasizes quality at each step
- Delivers frequent, tangible, working results
- Provides accurate and meaningful progress and status information with the minimum of overhead and disruption for the developers
- Is liked by clients, managers, and developers

Clients like FDD because they get to see real results early and progress reports written in terms that they understand.

Managers like FDD because it gives them a complete and accurate picture of project progress and status—the information they need to steer the project appropriately.

Developers like it because they get to work in small teams on small iterations and get to use the word *finished* frequently; they get to work on something new every few days. It involves them in analysis, design, and implementation. Analysis and design are not dictated by one or two elite analysts or architects. Developers also like FDD because it makes it easy to supply their managers with the status information they want with the minimum of effort and disruption.

Paul Szego, a Chief Programmer on the Singapore project, while signing author Steve Palmer's copy of the coloring book, wrote, "Who said software isn't fun? The best time I have had in a long time."

FDD answers positively the question, "What's in it for me?," for each role in the project. This contrasts with some processes used by managers to control developers because they feel the developers are not competent enough to be trusted; it also contrasts with processes used by developers to prevent themselves from being controlled because they do not feel their managers are competent enough to be trusted.

Since its introduction in the publication of the coloring book, FDD has proven effective within a growing number of development organizations. Its value is not restricted to salvaging projects in trouble; it is just as applicable to projects that are starting out, projects enhancing existing code, and projects tasked with creation of the second version of a system that was thrown together in a hurry last year.

FDD's underlying focus on producing frequent, tangible, working results at each level of the project (from developers to Chief Programmers, Chief Architects, Project Managers, Project Sponsors, and end users) makes it worthy of serious consideration by any software development organization that needs to deliver quality, business-critical software systems on time.

Those Responsible

Although a number of the Singapore project team members contributed to the development of FDD, there is no doubt that the main pair of villains involved were Jeff De Luca and Peter Coad.

Peter Coad is an industry-renowned object modeler, consultant, author, speaker, teacher, and wearer of Hawaiian shirts. Peter has designed hundreds of component and object systems, within nearly every industry imaginable.

Jeff De Luca, on the other hand, was schooled in the deep, dark arts of AS/400 operating system development at IBM's lonely laboratories in Rochester, Minnesota and is a highly successful, innovative, and experienced technical consultant and Project Manager. Pete has described him as "the best Project Manager I've ever worked with."

Why are they not writing this book? The simple answer is that they are both too busy building companies that build successful software applications. Jeff leads Nebulon Pty. Ltd. (www.nebulon.com) and is successfully using FDD to produce business systems for major clients in and around Melbourne, Australia. Pete is CEO of TogetherSoft (www.togethersoft.com), the company that produces the award-winning Together software development platform. All of the Unified Modeling Language (UML) diagrams in this book are constructed using TogetherSoft's Together ControlCenter software development platform.

Until now, people wanting information about FDD have been limited to the introduction in Chapter 6 of the coloring book [Coad 99], a little more discussion on the Nebulon Web site, and a few issues of the *Coad Letter* [Palmer], a monthly newsletter archived at www.togethercommunity.com and currently edited by Steve. The aims of this book are:

- To provide people with the in-depth information they have asked for about FDD
- To provide many practical hints and tips for applying FDD
- To discuss its development since the publication of the coloring book

If you are a developer or team leader, are involved in software development management in any way and need to make FDD work for you, or are simply interested in the whys and wherefores behind FDD, you are reading the right book.

Few, if any, of the ideas discussed are our own. Many of the ideas come from notes made by Jeff or Peter. Many of the ideas are inspired by the work of Fred Brooks [Brooks], Tom De Marco [De Marco], Luke Hohmann [Hohmann], and Gerald Weinberg [Weinberg 92, Weinberg 98]. In turn, much of their work builds on the work of Harlan Mills [Mills], Virginia Satir [Satir], Philip Crosby [Crosby], and others.

To attribute an idea to solely one person is naïveté; to attribute an idea solely to oneself is arrogance.

The principle of multiple minds taking a good idea and turning it into a great result pervades this book and FDD.

Reading This Book

The book is split into three sections:

1. Part 1 looks at some of the underlying principles and the important characteristics of any software development process and asks why it is so important. It then goes on to introduce the people, practices, and processes that comprise FDD. The section finishes with an introduction to tracking and reporting progress and work, and suggestions for report formats, design, and work packages.
2. Part 2 explores each process within FDD and provides extra hints, tips, and techniques to use when applying FDD.
3. Part 3 widens the discussion to answer the most frequently asked questions about FDD and to explore its impact on other important activities within a software development project that lie outside the process's targeted scope.

Making an Example

Writing a book about process is almost as mind-numbingly boring as reading one. In an attempt to keep you, the reader, awake until the end of the book, you'll find the authors suddenly bursting into role-play discussions at various points. We apologize, but it could be worse! Author Mac Felsing is a keen amateur opera singer; fortunately, printed media does not enable him to burst into song within these pages☺.

We use these role-play discussions to illustrate points within the context of a specific, small but realistic example project at Gary's Garage, a car dealership and servicing and repair shop.

Other books within this series are also using a similar example so that you can more readily compare the different approaches and processes described in the different books. We have chosen this example for two reasons:

1. It is large enough to present solid, real-world challenges.
2. We wanted to be able to present the contents of this book using a tangible, believable setting—one that you might relate to easily.

So, welcome to Gary's Garage. Gary owns the regional dealership franchise for a popular car manufacturer. At Gary's Garage, you can buy a new or quality used car, have your car serviced, and if unfortunately necessary, repaired. As the main dealer in the region, Gary also sells branded spare parts and accessories to other garages and to the general public. Gary and his team have volunteered to help specify and pilot a new sales and servicing management software system for the car manufacturer's dealerships.

Introducing...

The Development Team

- **Lisa, the Project Manager.** She is an experienced Project Manager for a small systems integrator and custom programming shop.
- **Steve, a Senior Developer.** He is experienced, intelligent, erudite, and British.
- **Mac, a Senior Developer.** He is just as experienced and intelligent as Steve but more handsome.
- **Various other extra staff as needed.** Don't you wish that staffing of projects was really this easy?

The Client Domain Experts

- **Mike the Mechanic.** As chief mechanic, Mike runs the servicing and repair workshop and a team of mechanics.
- **Stuart the Salesman.** He is in charge of the new and used car sales team and showrooms.
- **Sandy the Storeman (Parts Manager).** He is the parts manager, in charge of the spare parts inventory.
- **Rachel the Reception Manager.** She is in charge of scheduling services and repairs, and she handles customer relationships.
- **Anita the Accountant.** She runs a back office team, doing the books and payroll and managing the administrative side of the running a dealership franchise.

Steve: . . . Hey Mac, I'm told I'm going to be working with you on that car dealership project starting next week. What can you tell me about it?