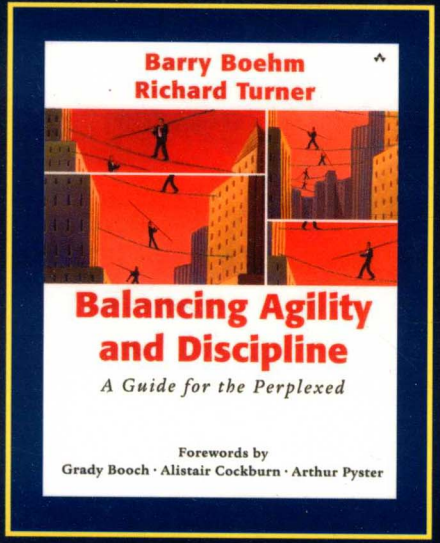
原 版 风 暴 系 列

PEARSON
Addison Wesley

Balancing Agility and Discipline
A Guide for the Perplexed

平衡敏捷和纪律
复杂软件系统开发指南
（影印版）

[美] Barry Boehm/Richard Turner 著
Booch/Cockburn/Pyster 序



- 美国国家工程院院士，AIAA、ACM、IEEE 会士作品
- 提供极具操作性的定义平衡软件开发策略的方法
- 来自现实的案例介绍，便于读者快速理解

中国电力出版社
www.infopower.com.cn

原 版 风 暴 系 列

Balancing Agility and Discipline
A Guide for the Perplexed

平衡敏捷和纪律
复杂软件系统开发指南
（影印版）

[美] Barry Boehm/Richard Turner 著
Booch/Cockburn/Pyster 序

中国电力出版社
www.infopower.com.cn

Balancing Agility and Discipline: A Guide for the Perplexed (ISBN 0-321-18612-5)
Boehm, Turner

图书在版编目（CIP）数据
平衡敏捷和纪律：复杂软件系统开发指南 /（美）勃姆（Boehm），（美）特纳（Turner）著.
—影印本. —北京：中国电力出版社，2004
（原版风暴系列）
ISBN 7-5083-2271-1
Ⅰ. 平... Ⅱ. ①勃... ② 特... Ⅲ. 软件开发－方法－英文 Ⅳ. TP311.52
中国版本图书馆 CIP 数据核字（2004）第 040316 号

丛　　名：原版风暴系列
书　　名：平衡敏捷和纪律：复杂软件系统开发指南（影印版）
编　　著：（美）Boehm, Turner
责任编辑：朱恩从
出版发行：中国电力出版社
　　地址：北京市三里河路6号　　邮政编码：100044
　　电话：（010）88515918　　传　　真：（010）88518169
印　　刷：北京丰源印刷厂
开　　本：787×1092　1/16　　印　　张：17.5
书　　号：ISBN 7-5083-2271-1
版　　次：2004 年 5 月北京第 1 版　　2004 年 5 月第 1 次印刷
定　　价：29.80 元

Foreword

by Grady Booch

There's a delightful irony in the fact that the very book you are holding in your hands has an agile pair of authors yet requires three times as many forewords as you'd find in any normal book.

Well, this is not a normal book; rather, it's a very pragmatic book that is not only quite approachable, but it is also immediately useful.

I have now personally lived through three generations of method wars. The era of structured analysis and design methods initially found its voice in methodologists such as Tom DeMarco, Ed Yourdon, Larry Constantine, Harlan Mills, Michael Jackson, and many others. There is an essential structured method that one can extract from their collective experience, but in the midst of that era, there was a veritable cacophony of competing approaches. The era of object-oriented analysis and design methods found its voice in methodologists such as Jim Rumbaugh, Ivar Jacobson, Peter Coad, Stephen Mellor, Watts Humphrey, myself, and many others. Here too one can extract some essential best practices (which is what the Rational Unified Process is all about), but still, that era was also characterized by dueling methods, each on a path to total world domination. Now we find ourselves in the post-dot-bomb era, and a fresh way of building systems has arisen, with individuals such as Kent Beck, Martin Fowler, Robert Martin, Jim Highsmith, and many others giving voice to the movement.

I expect that this won't be the last set of method wars I'll live through.

Actually, it's a sign of extreme health for our industry that there exists such a vibrant community of practice dealing with process and the developer experience. As I often quote from Bjarne Stroustrup, our civilization

runs on software. Building quality software that has economic value has been, is, and will remain a hard thing to do, and thus energy spent on improving processes is energy spent on reducing the friction of software development.

Barry and Rich are in an excellent position to examine the current method wars from a dispassionate, calculating way. In this book, they extract the essential practices from the more high-ceremony methods as well as the more low-ceremony ones. Their day in the life of the developer is absolutely wonderful in highlighting the differences and similarities among methods in this spectrum of ceremony.

This day in the life work alone is worth the price of this book, but they then go on to analyze two extended case studies from the real world. As they explain in the following section, taking a risk-driven approach is a pragmatic means of reconciling the strengths and weaknesses of disciplined and agile methods.

Being a certified bibliophile and a professional geek, I have more shelf space devoted to books on software methods than any reasonable human should possess. *Balancing Agility and Discipline* has a prominent place in that section of my library, because it has helped me sort through the noise and smoke of the current method wars.

—Grady Booch
Chief Scientist
IBM Rational Software

Foreword

by Alistair Cockburn

It was brave of Barry Boehm and Rich Turner to ask me to write a foreword for their book. They risk that as a founding agilite, I'll take exception to their characterization of the agile position.

Actually, I agree with them. They manage to peer through the rhetoric to uncover the strengths and weaknesses of the agile practices and to then compare and contrast those with the strengths and weaknesses of the plan-driven practices. They go further, showing how to borrow from each when the situation calls for it. This is no small accomplishment. I commend the authors for having managed it, and for making the result readable at the same time.

A word I find interesting throughout their discussion is *discipline.* The concept of discipline runs its separate way through both the plan-driven and agile camps. My Crystal Clear methodology is as low on the discipline scale as I can make it. On the other hand, eXtreme Programming (XP) calls for high levels of discipline, as anyone who has attempted it can attest. In fact, along with Watts Humphrey's Personal Software Process (PSP), I list XP as among the highest-discipline methodologies I know. So we have both low-discipline and high-discipline examples of agile approaches, and plan-driven and agile examples of high-discipline methodologies.

In their thoughtful way, Barry and Rich capture this and inform us that plan-driven and agile approaches lean on different meanings of the word *discipline:*

> [T]he term *disciplined,* whose dictionary definition includes both "common compliance with established processes" and "self-control,"

> is confined to "process compliance" by CMM bureaucrats, and confined to "self-control" by agile free spirits.

They remind us:

> If one has strong discipline without agility, the result is bureaucracy and stagnation. Agility without discipline is the unencumbered enthusiasm of a startup company before it has to turn a profit.

That is, both types of discipline are needed, in varying degrees. Part of the difference between plan-driven and agile approaches comes with highlighting one or the other meaning of the word *discipline.* Balancing your approach is much about balancing the two meanings of the word. That balancing is one of the things this book describes.

This is an outstanding book on an emotionally complicated topic. I applaud the authors for the care with which they have handled the subject.

—Alistair Cockburn
President, Humans and
Technology Project Director,
Agile Development Conference

Foreword

by Arthur Pyster

It is hard to argue against being agile and equally hard to disdain having discipline. The challenge is finding the right mix of agility and discipline. Many organizations have made great strides in productivity, predictability, quality, and cost using CMM-based process improvement—an approach that fosters disciplined processes. I have helped dozens of organizations make those strides using the Software CMM, the Systems Engineering CMM, and most recently, the CMM Integration. When properly applied, CMM-based process improvement works well. Of course, I have also seen organizations use the CMM to create stifling processes. Any tool can be misused.

For the past six years, I have worked at the Federal Aviation Administration (FAA)—the last four as Deputy Chief Information Officer. Billions of dollars are invested annually to safely move 700,000,000 passengers throughout U.S. airspace. The systems that manage air traffic share several characteristics that drive the FAA to disciplined execution of its development processes. Those systems require very high assurance and long lead times dictated by massive capital investment by government, airlines, manufacturers, and airports. System requirements are constrained by international agreements that ensure air traffic control works uniformly around the world. Air traffic control systems must be fair to all parties and must be installed while people are seven miles in the air. Careful long-range planning, stable requirements and architecture, and detailed documentation are essential to implementing and deploying such systems.

Nevertheless, processes for building air traffic systems can and do support aspects of agility. Ten years ago, air traffic control systems were built with very stilted processes. Today, spiral development, incremental

development, and incremental deployment are common. Lighter-weight processes are used early in the life cycle to prototype systems, refine requirements, and evolve architectures. Stakeholders are involved early and often to ensure that requirements are valid and human interfaces are effective. I expect the FAA to continue to probe where more agile processes can reduce cost and speed deployment, while recognizing the demanding environment in which these systems must operate.

Balancing agility and discipline is essential in any sizable project. The authors have done a commendable job of identifying five critical factors—personnel, criticality, size, culture, and dynamism—for creating the right balance of flexibility and structure. Their thoughtful analysis will help developers who must sort through the agile-discipline debate, giving them guidance to create the right mix for their projects.

—Arthur Pyster
Deputy Assistant Administrator
for Information Services and
Deputy Chief Information Officer
Federal Aviation Administration

Preface

Why We Wrote This Book

True believers represent software development alternatives

In the last few years, two ostensibly conflicting approaches to software development have competed for hegemony. Agile method supporters released a manifesto that shifts the focus from traditional plan-driven, process-based methods to lighter, more adaptive paradigms. Traditional methods have reasserted the need for strong process discipline and rigorous practices. True believers on both sides have raised strident, often antagonistic, voices.

This book is for the rest of us

We wrote this book for the rest of us—those caught in the middle of the method wars, simply trying to get our projects completed and accepted within too-tight schedules and budgets. We hope to clarify the perplexity about the roles of discipline, agility, and process in software development. We objectively compare and contrast the traditional, plan-driven approaches to the newer agile approaches and present an overview of their home grounds, strengths, and weaknesses. We then describe a risk-based approach to aid in balancing agility and discipline within a software development project.

Our goal is to help you in your business environment

We hope that this is a practical book. It is intended to be neither academic nor exhaustive, but pragmatic. It is based on our own development experiences, current and past literature, long conversations with proponents of agile and plan-driven approaches, teaching students how to balance discipline and agility, and years of observing and measuring software development in industry, government, and academia. We discuss the subject matter absent a need to choose sides. Our goal is to help you gain the understanding and information you need to integrate the approaches in a manner that best fits your business environment.

Who Should Read This Book

The perplexed—or just curious

This book is for perplexed software and management professionals who have heard the buzz about agile methods and want to separate the chaff from the wheat. Perhaps you have a CMM- or ISO-certified organization and want to know if and how agile methods can help you. Or perhaps some part of your organization has adopted agile methods and you are unsure of how they should fit in. Fundamentally, if you need to understand how the latest software development approaches can help meet business goals, this book is for you.

- *Software project managers and mid-level executives* should read this book to understand the agile/plan-driven controversy and learn how best to apply the new approaches in your organizations.
- *Software developers* should read this book to better understand how your field is evolving and what it means for your career.
- *Computer science and software engineering students* should read this book to better understand how to make choices about your own balance of agility and discipline, both in school and at work.
- *Academicians* should read this book to understand some of what your students are asking about, and how to help them make informed decisions.
- *Proponents of both agile and plan-driven methods* should read this book to dispassionately look at your opponent's ideas.
- *CIOs and CEOs* should read this book to help you understand what's going on in the software world and what implications it may have for your company.

How to Read This Book

Several ways to read the book

Most of you are busy people, and "must-read" material attacks you from all sides, 24/7. Some of you want to quickly assess the material for

later reflection. Others want to know how to implement the concepts we present. For that reason, we've tried to make this book easy to read quickly but with pointers to more in-depth material.

Margin summaries for fast track reading

To support the various reading needs, we've reused a format successfully employed by David Taylor in his outstanding *Object Technology: A Manager's Guide.* The margins contain a "fast track" summary of the text. We've included illustrations for key concepts. We've also included sidebar material that amplifies the text.

More information in the appendices

In order to meet the needs of the broadest possible audience, we have written the main text to provide basic information and relegated much of the technical material to appendices. Because of the authors' empirical backgrounds, one appendix covers the latest in empirical studies related to agility and discipline. The following icons will appear to indicate that additional material on the current topic is available in the appendices:

 Information on tools and techniques (Appendix D)

 Empirical material (Appendix E)

In a hurry? Use the fast track for a quick overview

If time is short, use the fast track summaries to scan the total content of the book, stopping to read things you find interesting or particularly applicable to your needs, and following the icons for specific technical information. If you find you need even more detailed material, see the References section for a list of additional resources.

First and last chapters are key

You can also tailor your reading through chapter selection. Reading the first and last chapters gives a pretty good idea of the material at a familiarization level. You can read the chapters in any order. Here is a quick summary:

Chapter 1 sets the stage for what follows. It introduces the main points and provides an executive summary of the book.

Chapter 2 compares the agile and plan-driven approaches and provides insight into the type of projects where each has been most successful—their home grounds.

Chapter 3 provides an experiential introduction to the approaches by describing how both a typical and not-so-typical day might be spent using each approach.

Chapter 4 presents two project case studies that illustrate the limits of pure agile and pure plan-driven implementations and the benefits of integrating the approaches.

Chapter 5 describes a risk-based approach for making methodology decisions that integrate agile and plan-driven practices, and illustrates it with representative examples.

Chapter 6 summarizes the material and offers some final observations.

Appendix A provides top-level descriptions of the major agile and plan-driven methods, highlighting their primary distinguishing factors, and a summary of those factors for comparison.

Appendices B through E provide technical and background information to support our analyses and speak to specific technical topics.

The Notes (listed by chapter) and the References follow Appendix E.

Acknowledgments

It is hard to know where to begin thanking the many people involved with creating this book. First there are the three foreword authors, who also reviewed the book draft and identified key improvements: Grady Booch, Alistair Cockburn, and Arthur Pyster. We were also fortunate to have a broad spectrum of reviewers of the full book draft whose perspectives provided many insightful improvement suggestions: Pekka Abrahamsson, Kristen Baldwin, Marguerite Brown, Scott Duncan, Peter Hantos, Denise Howard, Tony Jordano, Mikael Lindvall, Ken Schwaber, and Laurie Williams.

We would also like to thank very much those busy leaders in the agile and plan-driven communities who shared their time and expertise in helping us to see the software world from many different points of view: Scott Ambler, Ken Auer, Vic Basili, Kent Beck, Larry Bernstein, Winsor Brown, Bob Charette, Steve Cross, Michael Crowley, Christine Davis, Noopur Davis, Tom DeMarco, Nancy Eickelmann, Amr Elssamadisy, Hakan Erdogmus, Mike Falat, Martin Fowler, James Grenning, Jim Highsmith, Tom Hilburn, George Huling, Tuomo Kahkonen, Bil Kleb, William Krebs, Philippe Kruchten, Charles Leinbach, Wei Li, John Manzo, Frank Maurer, Granville Miller, Karen Owens, Mark Paulk, Gary Pollice, Dan Port, Don Reifer, Walker Royce, Gregory Schalliol, Kurt Schneider, Sarah Sheard, Giancarlo Succi, Roland Trauter, David Webb, Christian Wege, Laurie Williams, and William Wood. As the views of the people listed above differ considerably and we have tried to respect their views, our listing them does not imply that they agree with everything in the book.

We are especially grateful to our CeBASE colleagues—Mohammed Al-Said, LiGuo Huang, Apurva Jain, LaDonna Pierce, Meghna Shah,

Sachin Shah, Gunjan Sharman, and Paul Sitko from the University of Southern California Center for Software Engineering, and Patricia Costa, Atif Memon, Forrest Shull, Roseanne Tesoriero Tvedt, and Marv Zelkowitz from the Fraunhofer Center at the University of Maryland—for their support in the research of agile and plan-driven methods, particularly through the e-Workshops and the two agile methods workshops held at USC. Peter Gordon and the Addison-Wesley staff were, as always, excellent partners.

Finally, we offer love and thanks to our wives—Sharla and Jo—who not only put up with midnight faxes, travel extensions, and the general absent-mindedness that go with this kind of project, but also provided perceptive reviews and editorial suggestions that improved the book immensely. As always, they are our best inspiration and most honest critics, and we love them for both.

Prelude

Once upon a time, in a land steeped in metaphor, there lived an elephant. For many years, this reliable elephant served his village as the principal food gatherer and knew just what the village needed. He established paths through the jungle that always led him to the best roots, vegetables, nuts, and fruits. He knew which fruits he could reach with his trunk and which ones required some trunk shaking. His massive strength enabled him to bring back enough food for several days, so he always anticipated the requirements of the village and maintained adequate supplies. He was faithful to his task, was appreciated throughout the village, and thought his life most rewarding.

Alas, things began to change, as they often do in life and fable. The village cooks wanted different, rarer ingredients for their cooking, things the elephant had heard of but were not along his well-worn trail. He busily maintained stores of food that no one wanted but couldn't find time to make new paths for meeting new requests. The village grew impatient with the discouraged elephant, who just couldn't keep up with the demands.

Around the same time, there was a monkey in a nearby village whose job mirrored that of the elephant. Unlike the elephant, however, the agile monkey flitted across the jungle grabbing fruit as he saw it, finding the low-hanging fruit and bringing it quickly back to the village cooks. Rather than the time-proven trails of the elephant, the monkey relied on his memory and instincts

to find food and brought back only the amount needed that day. Sometimes he ran off looking for increasingly exotic foods and occasionally got lost. But his speed and agility always proved equal to the tasks the village set for him, and like the elephant, he was greatly appreciated.

Unfortunately, the monkey's life changed, too. His successful village grew larger every day. The monkey had so many requests that he was constantly on the move, trying to remember all the needs at every location. He had to make many more trips because he just didn't have the strength to carry everything requested at the same time. The village began to get impatient with him as well, and the monkey began to doubt he could do the job.

As luck would have it, the weary monkey and the discouraged elephant met one day. The monkey, trying to move quickly with a large load, noticed how much food the elephant was carrying in the panniers on his back. The elephant was impressed with the monkey's speed, how far he could travel, and how easily he could gather some of the food that the elephant struggled to reach. Both animals, proud of their skills, nevertheless acknowledged that there were obvious advantages to the other's abilities.

The elephant and monkey recognized the benefits of working together and decided to join forces. The monkey would use his agility to meet the new requests to find distant fruit, bringing it back to