

Barry Boehm
Richard Turner



Balancing Agility and Discipline

A Guide for the Perplexed

Forewords by

Grady Booch • Alistair Cockburn • Arthur Pyster

Balancing Agility and Discipline

Balancing Agility and Discipline: A Guide for the Perplexed, Portable Documents

Table of Contents

Contents

Foreword

Foreword

Foreword

Preface

Why We Wrote This Book

Who Should Read This Book

How to Read This Book

Acknowledgments

Prelude

Chapter 1 Discipline, Agility, and Perplexity

The Sources of Perplexity

Multiple Definitions

Distinguishing Method Use from Method Misuse

Overgeneralization Based on the Most Visible Instances

Claims of Universality

Early Success Stories

Purist Interpretations

Clarifying Perplexity

The Two Approaches

Table of Contents

Plan-Driven Methods

Agile Methods

Finding Middle Ground

Chapter 2 Contrasts and Home Grounds

Application Characteristics

Primary Goals

Size

Environment

Management Characteristics

Customer Relations

Planning and Control

Project Communication

Technical Characteristics

Requirements

Development

Testing

Personnel Characteristics

Customers

Developers

Culture

Summary

Home Grounds

Misconceptions

Five Critical Factors

Chapter 3 A Day in the Life

Typical Days

A Typical Day Using PSP/TSP

A Typical Day Using Extreme Programming

Table of Contents

Crisis Days

A Crisis Day with TSP/PSP

A Crisis Day with XP

Summary

Differences

Similarities

Observations

Chapter 4 Expanding the Home Grounds: Two Case Studies

Using Plans to Scale Up Agile Methods: Lease Management

Example

Assumption 1: The Effort to Develop or Modify a Story Does Not Increase
with Time and Story Number

Assumption 2: Trusting People to Get Everything Done on Time Is
Compatible with Fixed Schedules and Diseconomies of Scale

Assumption 3: Simple Design and YAGNI Scale Up Easily to Large Projects

Agile Methods Scaleup: Summing Up

Using Agility to Streamline Plan-Driven Methods: USAF/TRW

CCPDS-R Example

Individuals and Interactions over Processes and Tools: CCPDS-R

Working Software over Comprehensive Documentation: CCPDS-R

Customer Collaboration over Contract Negotiation: CCPDS-R

Responding to Change over Following a Plan: CCPDS-R

Summary

Chapter 5 Using Risk to Balance Agility and Discipline

An Overview of the Method

An Example Family of Applications: Agent-Based Planning Systems

An Intermediate Application: Supply Chain Management

Table of Contents

Step 1: SupplyChain.com Project Risk Ratings

Step 2: Compare the Agile and Plan-Driven Risks

Step 4a: Individual Risk Resolution Strategies

Step 4b: Risk-Based Strategy for SupplyChain.com System Development

Small Application: Event Planning

Step 1: Event Planning Project Risk Ratings

Step 2: Compare the Agile and Plan-Driven Risks

Steps 4a, 4b: Risk-Based Strategy for Event Planning System Development

Very Large Application: National Information System for Crisis Management (NISCM)

Step1: NISCM Project Risk Ratings

Step 2: Compare the Agile and Plan-Driven Risks

Steps 3 and 4: Risk-Based Strategy for NISCM System Development

Summary

Chapter 6 Conclusions

The Top Six Conclusions

No Agile or Plan-Driven Method Silver Bullet

Agile and Plan-Driven Method Home Grounds

Future Applications Will Need Both Agility and Discipline

Balanced Agility-Discipline Methods Are Emerging

Build Your Method UpDont Tailor It Down

Focus Less on MethodsMore on People, Values, Communication, and
Expectations Management

What Can You Do Next about Balancing Agility and Discipline?

Steps toward Balancing Software Development Agility and
Discipline

Afterword

Appendix A: Comparing the Methods

Table of Contents

Scrum

Thumbnail Sketch

Comments

References

Adaptive Software Development (ASD)

Thumbnail Sketch

Comments

References

Lean Development (LD)

Thumbnail Sketch

Comments

References

Crystal

Thumbnail Sketch

Comments

References

eXtreme Programming (XP)

Thumbnail Sketch

Comments

Reference

Dynamic Systems Development Method (DSDM)

Thumbnail Sketch

Comments

References

Rational Unified Process (RUP)

Thumbnail Sketch

Comments

References

Team Software Process (TSP)

Table of Contents

Thumbnail Sketch

Comments

References

Feature-Driven Development (FDD)

Thumbnail Sketch

Comments

References

Capability Maturity Model Integration (CMMI)

Thumbnail Sketch

Comments

References

Capability Maturity Model for Software (SW-CMM)

Thumbnail Sketch

Comments

References

Personal Software Process (PSP)

Thumbnail Sketch

Comments

References

Cleanroom

Thumbnail Sketch

Comments

References

Method Comparison Table

Appendix B: Manifesto for Agile Software Development

Principles behind the Agile Manifesto

Appendix C: Capability Maturity Models

A Short History of CMMs

Table of Contents

CMM Concepts

Using Models to Improve Processes

Appendix D: Tools for Balancing

D1. The Spiral Model Anchor Point Milestones

D2. Benefits Realization Analysis and the DMR Results Chain

Benefits Realized

Results Chain

D3. Schedule as an Independent Variable

Shared Vision and Expectations Management

Feature Prioritization

Schedule Range Estimation

Architecture and Core Capability Determination

Incremental Development

Change and Progress Monitoring and Control

Appendix E: Empirical Information

E1. The Cost of Change: Empirical Findings

E2. How Much Architecting Is Enough? A COCOMO II Analysis

E3. Experiments and Studies of Agile and Plan-Driven Methods

Overall Distribution of Project Size

Process Improvement

Team Software Process and Agile Methods

Pair Programming

Hybrid Agile/Plan-Driven Methods

Notes

References

Index