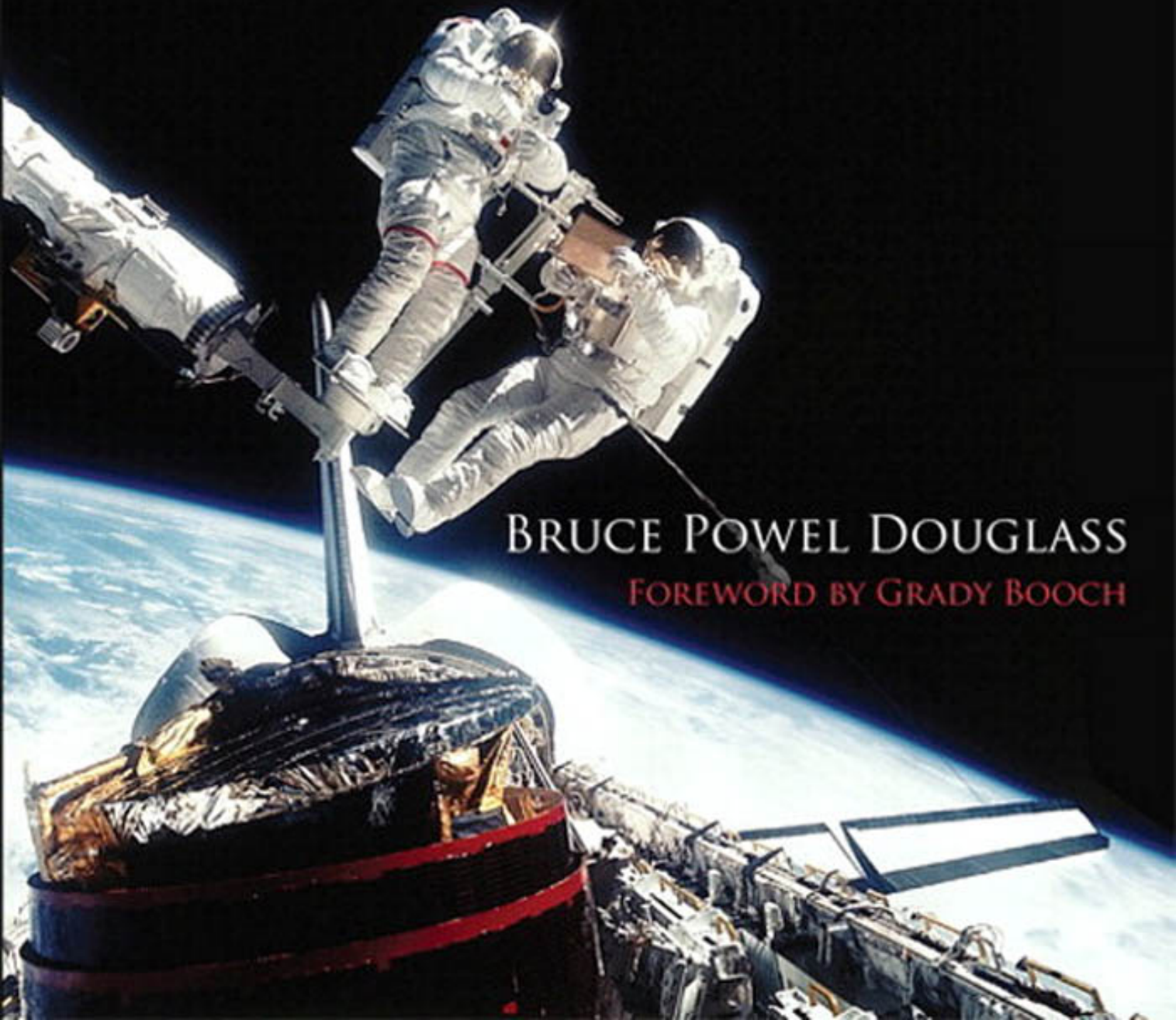
Real-Time Agility

The Harmony/ESW Method for Real-Time and Embedded Systems Development



Bruce Powel Douglass

Foreword by Grady Booch

Praise for *Real-Time Agility*

"Regardless of your perceptions of Agile, this is a must read! Douglass's book is a powerful and practical guide to a well-defined process that will enable engineers to confidently navigate the complexity, risk, and variability of real-time and embedded systems—including CMMI compliance. From requirements specification to product delivery, whatever your modeling and development environment, *this is the instruction manual.*"

—Mark Scoville, software architect

"This book will provide you with the framework of agile development for real-time projects ranging from embedded systems to web-based, data collection applications. I wish I had this book three years ago when we began a real-time, embedded drilling control system project, but all my engineers will be getting copies now that it is available. And, for my academic colleagues, this is the perfect book for graduate seminars in applied software development techniques."

—Don Shafer, chief technology officer, Athens Group;
adjunct professor, Cockrell School of Engineering,
The University of Texas at Austin

"We have used Dr. Douglass's books on real-time (*Doing Hard Time, Real-Time UML,* and *Real-Time Design Patterns*) for years. His books are always informative, accessible, and entertaining. *Real-Time Agility* continues that tradition, and I can't wait to introduce it to my colleagues."

—Chris Talbott, principal software designer

"Until now, agile software development has been mostly applied within the IT domain. This book breaks new ground by showing how to successfully traverse the perceived chasm between agility and real-time development. Although embedded systems impose challenging constraints on development teams, you can always benefit from increasing your agility."

—Scott W. Ambler, chief methodologist/Agile, IBM Rational;
author of Agile Modeling

Real-Time Agility: The Harmony/ESW Method for Real-Time and Embedded Systems Development

Table of Contents

Contents
Foreword
Preface
Acknowledgments
About the Author
Chapter 1: Introduction to Agile and Real-Time Concepts
- The Agile Manifesto
- Why Agile?
- Properties of Real-Time Embedded Systems
- Benefits of Agile Methods
  - Rapid Learning
  - Early Return on Investment
  - Satisfied Stakeholders
  - Improved Control
  - Responsiveness to Change
  - Earlier and Greater Reduction in Project Risk
  - Efficient High-Quality Development
- Agile Methods and Traditional Processes
  - Planning
  - Depth-First Development
  - Test-Driven Development

Pearson

Table of Contents

Embracing Change

Coming Up

Chapter 2: Concepts, Goals, and Benefits of Model-Driven Development

What Is MDA?

Why Model?

Key Concepts of MDA

Model

Metamodel

CIM

PIM

PSM

PSI

Model Transformation

MDA Technologies

MOF

UML

SysML

XMI

CWM

Profiles

Benefits of MDA

Portability

Reusability

Isolation from Technology Churn

Harmony's Five Key Architectural Views

Subsystem and Component Architecture

Concurrency and Resource Management Architecture

Distribution Architecture

Table of Contents

Safety and Reliability Architecture

Deployment Architecture

Secondary Architectural Views

Coming Up

Chapter 3: Harmony/ESW Principles and Practices

Harmony Core Principles

Your Primary Goal: Develop Working Software

Measure Progress against the Goal, Not the Implementation

Your Primary Measure of Progress Is Working Software

Don't Put Defects in Your Software!

Continuous Feedback Is Crucial

The Five Key Views of Architecture

Supplementing Your Architecture with Secondary Architectural Views

Plan, Track, and Adapt

The Leading Cause of Project Failure Is Ignoring Risk

Continuous Attention to Quality

Modeling Is Crucial

Optimizing the Right Things

Harmony Core Practices

Incrementally Construct

Use Dynamic Planning

Minimize Overall Complexity

Model with a Purpose

Use Frameworks

Prove the System under Development Is CorrectContinually

Create Software and Tests at the Same Time

Apply Patterns Intelligently

Manage Interfaces to Ease Integration

Use Model-Code Associativity

Table of Contents

- Coming Up

Chapter 4: Process Overview

- Why Process at All?
  - Harmony/ESW Process Introduction
- Harmony Time Frames
- Prototype-Based Spiral Development
- Harmony Macrocycle Process View
  - Prespiral Planning
  - Defining and Deploying the Development Environment
  - Developing Stakeholder Requirements
  - Controlling the Project
  - Change Management
- Harmony Spiral in Depth
  - Continuous Integration
  - Analysis
  - Design
  - Prepare for Validation
  - Model Review
  - Validation
  - Increment Review (Party Phase)
- What about Systems Engineering?
- What about CMMI?
- Combining Agile, MDA, and Harmony
- Coming Up

Chapter 5: Project Initiation

- What Do You Need to Get Started? The Baby Bear Plan
- Prespiral Planning
  - Creating the Schedule

Table of Contents

Creating the Team Structure

Planning for Reuse

Planning for Risk Reduction

Specifying the Logical Architecture

Performing the Initial Safety and Reliability Analysis

Developing Stakeholder Requirements

Defining the Product Vision

Finding and Outlining Stakeholder Requirements

Detailing the Stakeholder Requirements

Reviewing Stakeholder Requirements

Defining and Deploying the Development Environment

Tailoring the Development Process

Installing, Configuring, and Launching Development Tools

Installing and Configuring the CM Environment

Continuous Integration

What Is CM?

Continuous Configuration Management

Coming Up

Chapter 6: Agile Analysis

Prototype Definition

Plan the Iteration

Specifying the User Interface

Detailing the Use Cases

Generating System Requirements

Managing Safety and Reliability Requirements

Use Case White-Box Analysis

Use Case Consistency Analysis

Detailing System Requirements

Object Analysis

Table of Contents

Identifying Objects and Classes
Example from the Targeting Scanner Subsystem
Refining the Collaboration
Creating the Unit Test/Suite
Translating the Model into Code
Executing the Model
Factoring the Elements into the Model
Executing Unit Tests
Making the Change Set Available
Coming Up
Chapter 7: Agile Design
Optimization and the Use of Design Patterns
Design Patterns
Applying Design Patterns
Architectural Design
Primary and Secondary Architectural Views
Architectural Design Workflow
Optimizing Subsystem and Component Architecture
Optimizing Concurrency and Resource Management Architecture
Optimizing Distribution Architecture
Optimizing Safety and Reliability Architecture
Optimizing Deployment Architecture
Optimizing Secondary Architectural Views
Adding Architecture to the Transporter
Mechanistic Design
Mechanistic Design Workflow
Optimizing the Mechanistic Model
Practical Example: Optimizing the Collaboration
Detailed Design

Table of Contents

Detailed Design Workflow

Identifying "Special Needs" Classes

Optimizing Classes

Coming Up

Chapter 8: Agile Testing

Testing Concepts

Kinds of Test Cases

Levels of Testing

Test-Driven Development

Model-Based Testing

Testing Workflows

Unit Test

Unit Test Planning

Unit Test Execution

Integration Test

Continuous Integration

Managing Integration Tests

Validating and Accepting Changes to the Baseline

Making the Baseline Available

Validation Testing

Preparing for Validation

Validation

Coming Up

Chapter 9: Agile Process Optimization

Understanding Dynamic Planning

Tracking and Controlling

Controlling Project Workflow

Refining and Deploying the Development Environment

Table of Contents

Updating Risks

Updating the Schedule

Updating the Hazard Analysis

Change Management

Change Management Workflow

Model Reviews

The "Party Phase"

Party Phase Workflow

Issue Questionnaire

Reviewing Process

Reviewing the Schedule

Reviewing the Defect List and Defect Rates

Reviewing the Reuse Plan

Reviewing the Risk Management Plan

Summary

Appendix A: Starfleet ZX-1000 Transporter System Requirements Specification

1 Overview

2 Operational Modes

2.1 Cargo Transport Mode

2.2 Biomaterials Transport Mode

2.3 Personnel Transport Mode

2.4 Operational Submodes

2.5 Detoxification Submode

2.6 Biofilter Submode

3 General System Requirements

3.1 Timeliness Requirements

3.2 Power Requirements

3.3 Target Scan and Coordinate Lock

Table of Contents

3.4 Dematerialization

3.5 Pattern Storage

3.6 Doppler Compensations

3.7 Matter Stream Transmission

3.8 Rematerialization

3.9 Operator Control

3.10 Transportation Sequencing

3.11 System Configuration

4 Major System Components

4.1 Operator Console

4.2 Transport Chamber

4.3 Transport Sequencer

4.4 Primary and Secondary Energizing Coils

4.5 Phase Transition Coils

4.6 Quantum Molecular Imaging Scanners

4.7 Pattern Buffer

4.8 Biofilter

4.9 Emitter Array

4.10 Targeting Scanners

5 Secondary Functions

5.1 Site-to-Site Transport

5.2 Pattern Buffer Storage

5.3 Hazardous Material Dispersal

5.4 Near-Warp Transport

5.5 Warp Transport

5.6 Life Sign Scanner

5.7 Biofilter

5.8 Hazardous Materials Filter

5.9 Phase/Frequency/Packet Compensation

Appendix B: Harmony/ESW and CMMI: Achieving

Table of Contents

- Compliance
- Abstract
- CMMI Basics
  - Process Areas
  - Maturity Levels
- Achieving CMMI Compliance with Harmony/ESW
  - Maturity Level 1
  - Maturity Level 2
  - Maturity Level 3
  - Maturity Level 4
  - Maturity Level 5
- Summary
- Further Reading

Glossary

- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- L
- M
- N
- O

Table of Contents

P
Q
R
S
T
U
V
W
X
Index