



THE PROCESS OF

SOFTWARE ARCHITECTING

PETER EELES
PETER CRIPPS

FOREWORD BY GRADY BOOCH



Praise for *The Process of Software Architecting*

“The role of software architect has really come into its own in the last few years and is now recognized as a key determinant of project success. However, even today there is little common understanding of how to do the job: analyze requirements, understand concerns, evaluate alternatives, and construct and document an architectural description that is fit for purpose. Eeles and Cripps have filled this gap in their useful and practical book. The material is clearly and simply laid out, and follows a logical progression from inception through delivery, with tasks and work products clearly explained and illustrated using a real-world case study. This is a vital handbook for everyone from brand new architects to seasoned professionals.”

—Nick Rozanski, *coauthor of Software Systems Architecture*

“If you need a thorough and authoritative reference for a complete software architecture process, then look no further. Peter Eeles and Peter Cripps have produced a definitive guide and reference to just such a process. Being precisely defined using a metamodel, illustrated with a realistic case study, and clearly related to key standards such as UML, RUP, and IEEE 1471, the process presented in this book provides a valuable guide for those developing software architectures for large projects. I have no doubt that it will become a valued reference for many software architects.”

—Eoin Woods, *coauthor of Software Systems Architecture*

“Eeles and Cripps distill years of experience into a single guide that helps the reader understand not just what architects produce, but *how* they produce it. *The Process of Software Architecting* is a very practical guide filled with lessons learned and pitfalls to avoid. Practicing architects will want to refer to it as they hone their skills, and aspiring architects will gain key insights that otherwise could take painful years of experience to acquire.”

—Bob Kitzberger, *program director, Strategy, IBM Software Group*

“For most of my career in this field, software architecture has had the feel of being a black art that only a select few gurus and wizards had a talent for. This book follows on from industry best practice and a wealth of author experience to bring solutions architecture into the realms of being a true engineering discipline. Now I have a real guide I can pass on to new practitioners, a guide that embodies what used to require years of trial and error.”

—Colin Renouf, *enterprise architect and technology author,
vice chairman, WebSphere User Group, UK*

Process of Software Architecting, The

Table of Contents

Contents

Foreword

Preface

Acknowledgments

About the Authors

Chapter 1 Introduction

- Applying the Process

- The Process in Brief

- Scope

- Summary

Chapter 2 Architecture, Architect, Architecting

- Architecture

 - An Architecture Defines Structure

 - An Architecture Defines Behavior

 - An Architecture Focuses on Significant Elements

 - An Architecture Balances Stakeholder Needs

 - An Architecture Embodies Decisions Based on Rationale

 - An Architecture May Conform to an Architectural Style

 - An Architecture Is Influenced by Its Environment

 - An Architecture Influences Development Team Structure

 - An Architecture Is Present in Every System

 - An Architecture Has a Particular Scope

- Architect

Table of Contents

- The Architect Is a Technical Leader
- The Architect Role May Be Fulfilled by a Team
- The Architect Understands the Software Development Process
- The Architect Has Knowledge of the Business Domain
- The Architect Has Technology Knowledge
- The Architect Has Design Skills
- The Architect Has Programming Skills
- The Architect Is a Good Communicator
- The Architect Makes Decisions
- The Architect Is Aware of Organizational Politics
- The Architect Is a Negotiator

Architecting

- Architecting Is a Science
- Architecting Is an Art
- Architecting Spans Many Disciplines
- Architecting Is an Ongoing Activity
- Architecting Is Driven by Many Stakeholders
- Architecting Often Involves Making Trade-Offs
- Architecting Acknowledges Experience
- Architecting Is Both Top-Down and Bottom-Up

The Benefits of Architecting

- Architecting Addresses System Qualities
- Architecting Drives Consensus
- Architecting Supports the Planning Process
- Architecting Drives Architectural Integrity
- Architecting Helps Manage Complexity
- Architecting Provides a Basis for Reuse
- Architecting Reduces Maintenance Costs
- Architecting Supports Impact Analysis

Table of Contents

Summary

Chapter 3 Method Fundamentals

Key Concepts

Method Content

Role

Work Product

Activity

Task

Process

Waterfall Processes

Iterative Processes

Agile Processes

Summary

Chapter 4 Documenting a Software Architecture

The End Game

Key Concepts

Viewpoints and Views

Basic Viewpoints

Cross-Cutting Viewpoints

Views and Diagrams

Benefits of Viewpoints and Views

Models

Levels of Realization

Benefits of Models

Characteristics of an Architecture Description Framework

The 4 + 1 View Model of Software Architecture

Zachman Framework

Table of Contents

Rozanski and Woods

An Architecture Description Framework

Viewpoints

Work Products

Levels of Realization

View Correspondence

The Software Architecture Document

Summary

Chapter 5 Reusable Architecture Assets

Sources of Architecture

An Architecture Asset Metamodel

Development-Time Assets

Run-Time Assets

Asset Types

Reference Architecture

Development Method

Viewpoint Catalog

Architectural Style

Architectural Mechanism

Pattern

Reference Model

Architecture Decision

Existing Application

Packaged Application

Application Framework

Component Library/Component

Attributes of an Architecture Asset

Other Reuse Considerations

Table of Contents

Summary

Chapter 6 Introduction to the Case Study

Applying the Process

Scope of the Case Study

The Project Team

External Influences

Application Overview

The YourTour Vision

Problem Statement

Stakeholders

Functionality

Qualities

Constraints

Summary

Chapter 7 Defining the Requirements

Relating Requirements to Architecture

Functional and Non-Functional Requirements

Techniques for Documenting Requirements

Applying the Process

Understanding the Task Descriptions

Define Requirements: Activity Overview

Task: Collect Stakeholder Requests

Task: Capture Common Vocabulary

Task: Define System Context

Task: Outline Functional Requirements

Task: Outline Non-Functional Requirements

Task: Prioritize Requirements

Table of Contents

Task: Detail Functional Requirements

Task: Detail Non-Functional Requirements

Task: Update Software Architecture Document

Task: Review Requirements with Stakeholders

Summary

Chapter 8 Creating the Logical Architecture

Moving from Requirements to Solution

How Much Logical Architecture?

Minimizing Logical Architecture

Logical Architecture as an Investment

The Importance of Traceability

Applying the Process

Create Logical Architecture: Activity Overview

Task: Survey Architecture Assets

Task: Define Architecture Overview

Task: Document Architecture Decisions

Task: Outline Functional Elements

Task: Outline Deployment Elements

Task: Verify Architecture

Task: Build Architecture Proof-of-Concept

Task: Detail Functional Elements

Task: Detail Deployment Elements

Task: Validate Architecture

Task: Update Software Architecture Document

Task: Review Architecture with Stakeholders

Summary

Table of Contents

Chapter 9 Creating the Physical Architecture

Moving from Logical to Physical Architecture

Applying the Process

Creating the Physical Architecture: Activity Overview

Task: Survey Architecture Assets

Task: Define Architecture Overview

Task: Document Architecture Decisions

Task: Outline Functional Elements

Mapping Logical Functional Elements to Physical Functional Elements

Identifying Physical Functional Elements

Procuring Products

Accommodating Technology-Specific Patterns

Task: Outline Deployment Elements

Mapping Logical Deployment Elements to Physical Deployment Elements

Identifying Physical Deployment Elements

Procuring Hardware

Task: Verify Architecture

Task: Build Architecture Proof-of-Concept

Task: Detail Functional Elements

Task: Detail Deployment Elements

Task: Validate Architecture

Task: Update Software Architecture Document

Task: Review Architecture with Stakeholders

Summary

Chapter 10 Beyond the Basics

The Architect and the Project Team

The Architect and Requirements

Table of Contents

The Architect and Development

The Architect and Test

The Architect and Project Management

The Architect and Configuration Management

The Architect and Change Management

The Architect and the Development Environment

The Architect and Business Analysis

The Architect and External Influences

Enterprise Architecture

Design Authority

Infrastructure Provider

Application Maintenance Provider

Architecting Complex Systems

Many Distinct Functions Are Being Developed

Many People Are Involved in Development

The System Is Highly Distributed

The Development Team Is Distributed

Operational Qualities Are Extremely Challenging

There Is a System of Systems

Summary

In Conclusion: A Note from the Authors

Appendix A: Software Architecture Metamodel

Definition of Metamodel Terms

Appendix B: Viewpoint Catalog

Stakeholder Summary

Basic Viewpoints

Cross-Cutting Viewpoints

View Correspondence

Table of Contents

Appendix C: Method Summary

- Roles

- Work Products

- Activities

- Tasks

 - Activity: Define Requirements

 - Activity: Create Logical Architecture

 - Activity: Create Physical Architecture

- Phases

 - Inception

 - Elaboration

 - Construction

 - Transition

Appendix D: Architectural Requirement Checklist

- Functional Requirements

- Usability Requirements

- Reliability Requirements

- Performance Requirements

- Supportability Requirements

- Constraints

 - Business Constraints

 - Architecture Constraints

 - Development Constraints

 - Physical Constraints

Glossary

- A

- B

Table of Contents

C

D

E

F

I

M

N

P

Q

R

S

T

V

W

References

Index