

# Managing Software Requirements

## Foreword

The Rock Problem

About This Book

## Preface

Context and Acknowledgments

Requirements Lessons from Building Software for Others

Lessons from Building High-Assurance Systems

Lessons from the Requirements Management Business

Experiences at Rational Software

Summary

## I: Introduction

### 1. The Requirements Problem

The Goal

A Look at the Data

Root Causes of Project Success and Failure

### 2. Introduction to Requirements Management

Definitions

Application of Requirements Management Techniques

The Road Map

Summary

### 3. The Software Team

Software Development as a Team Activity

The Case Study

Summary

## II: Team Skill 1: Analyzing the Problem

### 4. The Five Steps in Problem Analysis

Step 1: Gain Agreement on the Problem Definition

Step 2: Understand the Root Causes—The Problem Behind the Problem

Step 3: Identify the Stakeholders and the Users

Step 4: Define the Solution System Boundary

Step 5: Identify the Constraints to Be Imposed on the Solution

Summary

Looking Ahead

### 5. Business Modeling

Purpose of Business Modeling

Using Software Engineering Techniques for Business Modeling

From the Business Models to the Systems Model

When to Use Business Modeling

Summary

Looking Ahead

## 6. Systems Engineering of Software-Intensive Systems

What Is Systems Engineering?

Requirements Allocation in Systems Engineering

The Case Study

Team Skill 1 Summary

## III: Team Skill 2: Understanding User Needs

### 7. The Challenge of Requirements Elicitation

Barriers to Elicitation

Techniques for Requirements Elicitation

### 8. The Features of a Product or System

Stakeholder and User Needs

Features

### 9. Interviewing

The Interview Context

Value-Added Context

The Moment of Truth: The Interview

Compiling the Need Data

A Note on Questionnaires

### 10. Requirements Workshops

Accelerating the Decision Process

Preparing for the Workshop

Role of the Facilitator

Setting the Agenda

Running the Workshop

### 11. Brainstorming and Idea Reduction

Live Brainstorming

Idea Reduction

Web-Based Brainstorming

The Case Study: The HOLIS 2000 Requirements Workshop

### 12. Storyboarding

Types of Storyboards

What Storyboards Do

Tools and Techniques for Storyboarding

Tips for Storyboarding

Summary

### 13. Applying Use Cases

Building the Use-Case Model

Applying Use Cases to Requirements Elicitation

Case Study: The Use Cases for HOLIS

Summary

### 14. Role Playing

How to Role Play

Techniques Similar to Role Playing

Summary

### 15. Prototyping

Types of Prototypes

[Requirements Prototypes](#)

[What to Prototype](#)

[Building the Prototype](#)

[Evaluating the Results](#)

[Summary](#)

[Team Skill 2 Summary](#)

#### [IV: Team Skill 3: Defining the System](#)

##### [16. Organizing Requirements Information](#)

[Organizing Requirements of Complex Hardware and Software Systems](#)

[Organizing Requirements for Product Families](#)

[On "Future" Requirements](#)

[Business and Marketing Requirements versus Product Requirements](#)

[The Case Study](#)

[Summary](#)

##### [17. The Vision Document](#)

[Components of the Vision Document](#)

[The "Delta Vision" Document](#)

##### [18. The Champion](#)

[The Role of the Product Champion](#)

[The Product Champion in a Software Product Environment](#)

[The Product Champion in an IS/IT Shop](#)

[Team Skill 3 Summary](#)

#### [V: Team Skill 4: Managing Scope](#)

##### [19. The Problem of Project Scope](#)

[Components of Project Scope](#)

[The Hard Question](#)

##### [20. Establishing Project Scope](#)

[The Requirements Baseline](#)

[Setting Priorities](#)

[Assessing Effort](#)

[Adding the Risk Element](#)

[Reducing Scope](#)

[The Case Study](#)

##### [21. Managing Your Customer](#)

[Engaging Customers to Manage Their Project Scope](#)

[Communicating the Result](#)

[Negotiating with the Customer](#)

[Managing the Baseline](#)

##### [22. Scope Management and Software Development Process Models](#)

[The Waterfall Model](#)

[The Spiral Model](#)

[The Iterative Approach](#)

[What to Do, What to Do ...](#)

[Team Skill 4 Summary](#)

#### [VI: Team Skill 5: Refining the System Definition](#)

## 23. Software Requirements

Definition of Software Requirements

Relationship between Features and Software Requirements

The Requirements Dilemma: What versus How

More on Requirements versus Design

A Further Characterization of Requirements

Using Parent-Child Requirements to Increase Specificity

Looking Ahead

## 24. Refining the Use Cases

Questions to Ask

Refining Use-Case Specifications

The Case Study: Anatomy of a Simple Use Case

Looking Ahead

## 25. A Modern Software Requirements Specification

The Modern SRS Package

Documenting Functional Requirements

Looking Ahead

## 26. On Ambiguity and Specificity

Finding the "Sweet Spot"

Mary Had a Little Lamb

Techniques for Disambiguation

What to Do?

## 27. Quality Measures of Software Requirements

Nine Quality Measures

Quality Measures for the Use-Case Model

Quality Measures of the Modern SRS Package

## 28. Technical Methods for Specifying Requirements

Pseudocode

Finite State Machines

Decision Trees and Decision Tables

Graphical Decision Trees

Activity Diagrams

Entity-Relationship Models

Object-Oriented Modeling

Data Flow Diagrams

Maintenance of Specifications

Case Study

Team Skill 5 Summary

## VII: Team Skill 6: Building the Right System

### 29. Building the Right System Right: Overview

Continually Confirm that the Development Is on Track

Confirm that the Development Results Are Correct

Learn How to Cope with Change that Occurs during the Development Process

Looking Ahead

### 30. From Requirements to Implementation

Mapping Requirements to Design and Code

Realizing Use Cases in the Design Model

From Design to Implementation

[Summary](#)  
[Looking Ahead](#)

### [31. Using Traceability to Support Verification](#)

[The Role of Traceability in Requirements Verification](#)  
[Using Traceability Tools](#)  
[Proceeding without Traceability Tools](#)  
[Thinking about Verification and Traceability](#)  
[Looking Ahead](#)

### [32. Validating the System](#)

[Validation](#)  
[Case Study: Testing Use Cases](#)  
[Testing Discrete Requirements](#)  
[Testing Design Constraints](#)  
[Looking Ahead](#)

### [33. Using ROI to Determine the V&V Effort](#)

[Depth versus Coverage](#)  
[What to Verify and Validate](#)  
[Looking Ahead](#)

### [34. Managing Change](#)

[Why Do Requirements Change?](#)  
["We Have Met the Enemy, and They Is Us"](#)  
[A Process for Managing Change](#)  
[Requirements Configuration Management](#)  
[Summary](#)  
[Team Skill 6 Summary](#)

### [35. Getting Started](#)

[Dedication](#)  
[What We've Learned So Far](#)  
[Your Prescription for Requirements Management](#)  
[Now, On to the Next Release!](#)

#### [A. HOLIS Artifacts](#)

[Background of the Case Study](#)  
[Team Skill 1: Analyzing the Problem](#)  
[Team Skill 2: Understanding User Needs](#)  
[Team Skill 3: Defining the System](#)  
[Team Skill 4: Managing Scope](#)  
[Team Skill 5: Refining the System Definition](#)  
[Team Skill 6: Building the Right System](#)

#### [B. Vision Document Template](#)

[Table of Contents](#)

#### [C. Modern SRS Package Template](#)

#### [D. Requirements Management in the SEI-CMM and within ISO 9000](#)

[Requirements Management in SEI-CMM](#)  
[Requirements Management in ISO 9000](#)

#### [E. Requirements Management in the Rational Unified Process](#)

[Structure of the Rational Unified Process](#)

Requirements Management in the Rational Unified Process  
Process Integration

Bibliography

--