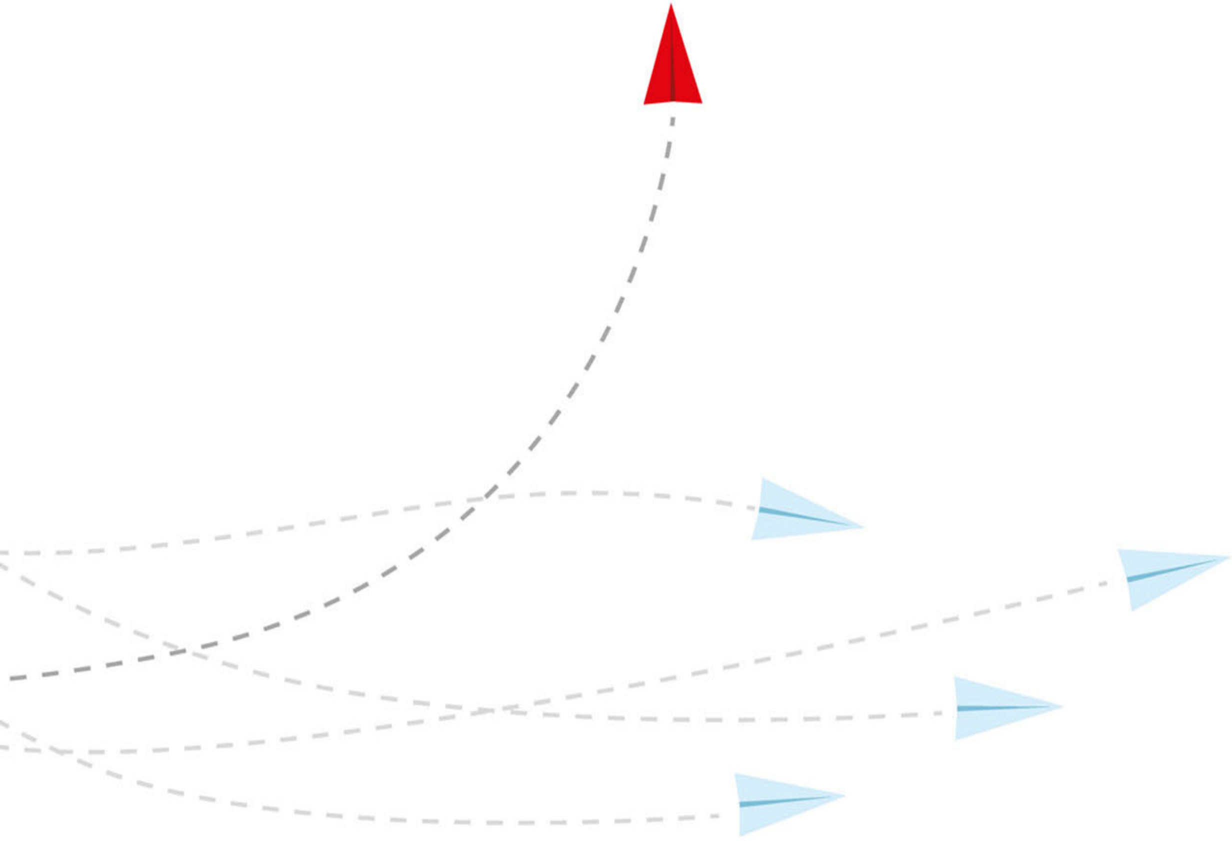
RIGHTING SOFTWARE

A Method for System and Project Design



JUVAL LÖWY

Praise for Juval Löwy

"I attended both the Architect's Master Class and the Project Design Master Class. Before these two classes I had almost lost all hope of ever being able to figure out why the efforts of my team were never leading to a successful end, and I was struggling to find a working solution to stop the insane death march we were on. The Master Classes opened my eyes to a world where software development is elevated to the level of all other engineering disciplines and is conducted in a professional, predictable, and reliable manner, resulting in high-quality working software developed on time and within budget. The knowledge gained is priceless! From revealing how to create a solid and sound architecture, which withstands ever-changing user requirements, to the intricate details on how to plan and guide the project to a successful end—all this was presented with expertise and professionalism that are hard to match. Considering that every bit of distilled truth Juval shared with us in class is acquired, tested, and proven in real life, it transforms this learning experience into a powerful body of knowledge that is an absolute necessity for anyone who aspires to be a Software Architect."

—Rossen Totev, software architect/project lead

"The Project Design Master Class is a career-changing event. Having come from an environment where deadlines and budgets are almost pathologically abused, having the opportunity to learn from Juval was a godsend. Piece by piece he provided the parts and the appropriate tools for properly designing a project. The result is that costs and timelines are kept in check in the dynamic and even chaotic environment of modern software development. Juval says that you are going to engage in asymmetric warfare against overdue and over cost, and you walk away truly feeling that you have a gun to take to a knife fight. There is no magic—only the application of basic engineering and manufacturing tenets to software—but you will go back to your office feeling like a wizard."

—Matt Robold, software development manager, West Covina Service Group

"Fantastic experience. Changed my way of thinking on how to approach software development. I always knew some of what I was thinking was right with regard to design and coding. I never could express it in words but now I have them. It not only affects my way of thinking about software design but also other types of design."

—Lee Messick, lead architect

Righting Software

Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Preface

About the Author

Chapter 1 The Method

- What Is The Method?
  - Design Validation
  - Time Crunch
  - Eliminating Analysis-Paralysis
  - Communication
- What The Method Is Not

PART I: SYSTEM DESIGN

- Chapter 2 Decomposition
  - Avoid Functional Decomposition
    - Problems with Functional Decomposition
    - Reflecting on Functional Decomposition
    - Avoid Domain Decomposition
    - Faulty Motivation
    - Testability and Design
    - Example: Functional Trading System
  - Volatility-Based Decomposition

Pearson

Table of Contents

Decomposition, Maintenance, and Development

Universal Principle

Volatility-Based Decomposition and Testing

The Volatility Challenge

Identifying Volatility

Volatile versus Variable

Axes of Volatility

Solutions Masquerading as Requirements

Volatilities List

Example: Volatility-Based Trading System

Resist the Siren Song

Volatility and the Business

Design for Your Competitors

Volatility and Longevity

The Importance of Practicing

Chapter 3 Structure

Use Cases and Requirements

Required Behaviors

Layered Approach

Using Services

Typical Layers

The Client Layer

The Business Logic Layer

The ResourceAccess Layer

The Resource Layer

Utilities Bar

Classification Guidelines

Whats in a Name

The Four Questions

Managers-to-Engines Ratio

Key Observations

Subsystems and Services

Table of Contents

Incremental Construction

About Microservices

Open and Closed Architectures

Open Architecture

Closed Architecture

Semi-Closed/Semi-Open Architecture

Relaxing the Rules

Design Donts

Strive for Symmetry

Chapter 4 Composition

Requirements and Changes

Resenting Change

Design Prime Directive

Composable Design

Core Use Cases

The Architects Mission

There Is No Feature

Handling Change

Containing the Change

Chapter 5 System Design Example

System Overview

Legacy System

New System

The Company

Use Cases

The Anti-Design Effort

The Monolith

Granular Building Blocks

Domain Decomposition

Business Alignment

The Vision

The Business Objectives

Table of Contents

        - Mission Statement
    - The Architecture
        - TradeMe Glossary
        - TradeMe Areas of Volatility
        - Static Architecture
        - Operational Concepts
        - Workflow Manager
    - Design Validation
        - Add Tradesman/Contractor Use Case
        - Request Tradesman Use Case
        - Match Tradesman Use Case
        - Assign Tradesman Use Case
        - Terminate Tradesman Use Case
        - Pay Tradesman Use Case
        - Create Project Use Case
        - Close Project Use Case
    - Whats Next?

PART II: PROJECT DESIGN

- Chapter 6 Motivation
    - Why Project Design?
    - Project Design and Project Sanity
    - Assembly Instructions
    - Hierarchy of Needs
- Chapter 7 Project Design Overview
    - Defining Success
        - Reporting Success
    - Project Initial Staffing
        - Architect, Not Architects
        - The Core Team
    - Educated Decisions
        - Plans, Not Plan

Pearson

Table of Contents

Software Development Plan Review

Services and Developers

Design and Team Efficiency

Task Continuity

Effort Estimations

Classic Mistakes

Estimation Techniques

Overall Project Estimation

Activity Estimations

Critical Path Analysis

Project Network

The Critical Path

Assigning Resources

Scheduling Activities

Staffing Distribution

Project Cost

Project Efficiency

Earned Value Planning

Classic Mistakes

The Shallow S Curve

Roles and Responsibilities

Chapter 8 Network and Float

The Network Diagram

The Node Diagram

The Arrow Diagram

Arrow versus Node Diagrams

Floats

Total Float

Free Float

Calculating Floats

Visualizing Floats

Floats-Based Scheduling

Table of Contents

Float and Risk

Chapter 9 Time and Cost

Accelerating Software Projects

Schedule Compression

Working with Better Resources

Working in Parallel

Parallel Work and Cost

TimeCost Curve

Points on TimeCost Curve

Discrete Modeling

Avoiding Classic Mistakes

Project Feasibility

Finding Normal Solutions

Project Cost Elements

Direct Cost

Indirect Cost

Accounting versus Value

Total, Direct, and Indirect Costs

Compression and Cost Elements

Staffing and Cost Elements

Fixed Cost

Network Compression

Compression Flow

Chapter 10 Risk

Choosing Options

TimeRisk Curve

Actual TimeRisk Curve

Risk Modeling

Normalizing Risk

Risk and Floats

Risk and Direct Cost

Criticality Risk

Pearson

Table of Contents

Fibonacci Risk

Activity Risk

Criticality versus Activity Risk

Compression and Risk

Execution Risk

Risk Decompression

How to Decompress

Decompression Target

Risk Metrics

Chapter 11 Project Design in Action

The Mission

The Static Architecture

The Call Chains

List of Activities

Network Diagram

Planning Assumptions

Finding the Normal Solution

Unlimited Resources (Iteration 1)

Network and Resource Problems

Infrastructure First (Iteration 2)

Limited Resources

Going Subcritical (Iteration 7)

Choosing the Normal Solution

Network Compression

Compression Using Better Resources

Introducing Parallel Work

End of Compression Iterations

Throughput Analysis

Efficiency Analysis

TimeCost Curve

TimeCost Correlation Models

The Death Zone

Table of Contents

Planning and Risk

Risk Decompression

Rebuilding the TimeCost Curve

Modeling Risk

Risk Inclusion and Exclusion

SDP Review

Presenting the Options

Chapter 12 Advanced Techniques

God Activities

Handling God Activities

Risk Crossover Point

Deriving the Crossover Point

Finding the Decompression Target

Geometric Risk

Geometric Criticality Risk

Geometric Fibonacci Risk

Geometric Activity Risk

Geometric Risk Behavior

Execution Complexity

Cyclomatic Complexity

Project Type and Complexity

Compression and Complexity

Very Large Projects

Complex Systems and Fragility

Network of Networks

Designing a Network of Networks

Small Projects

Design by Layers

Pros and Cons

Layers and Construction

Chapter 13 Project Design Example

Table of Contents

- Estimations
    - Individual Activity Estimations
    - Overall Project Estimation
- Dependencies and Project Network
    - Behavioral Dependencies
    - Nonbehavioral Dependencies
    - Overriding Some Dependencies
    - Sanity Checks
- The Normal Solution
    - Network Diagram
    - Planned Progress
    - Planned Staffing Distribution
    - Cost and Efficiency
    - Results Summary
- Compressed Solution
    - Adding Enabling Activities
    - Assigning Resources
    - Planned Progress
    - Planned Staffing Distribution
    - Cost and Efficiency
    - Results Summary
- Design by Layers
    - Design by Layers and Risk
    - Staffing Distribution
    - Results Summary
- Subcritical Solution
    - Duration, Planned Progress, and Risk
    - Cost and Efficiency
    - Results Summary
- Comparing the Options
- Planning and Risk
    - Risk Decompression
    - Recalculating Cost

Table of Contents

Preparing for the SDP Review

Chapter 14 Concluding Thoughts

When to Design a Project

The Real Answer

Getting Ahead in Life

General Guidelines

Architecture versus Estimations

Design Stance

Optionality

Compression

Planning and Risk

Design of Project Design

In Perspective

Subsystems and the Timeline

The Hand-O?

Junior Hand-O?

Senior Hand-O?

Senior Developers as Junior Architects

In Practice

Debrie? ng Project Design

About Quality

Quality-Control Activities

Quality-Assurance Activities

Quality and Culture

Appendix A Project Tracking

Activity Life Cycle and Status

Phase Exit Criteria

Phase Weight

Activity Status

Project Status

Progress and Earned Value

Pearson

Table of Contents

    - Accumulated E? ort
    - Accumulated Indirect Cost
  - Tracking Progress and E? ort
  - Projections
  - Projections and Corrective Actions
    - All Is Well
    - Underestimating
    - Resource Leak
    - Overestimating
  - More on Projections
    - The Essence of a Project
    - Handling Scope Creep
    - Building Trust
- Appendix B Service Contract Design
  - Is This a Good Design?
  - Modularity and Cost
    - Cost per Service
    - Integration Cost
    - Area of Minimum Cost
  - Services and Contracts
    - Contracts as Facets
    - From Service Design to Contract Design
    - Attributes of Good Contracts
  - Factoring Contracts
    - Design Example
    - Factoring Down
    - Factoring Sideways
    - Factoring Up

Table of Contents

- Contract Design Metrics
  - Measuring Contracts
  - Size Metrics
  - Avoid Properties
  - Limit the Number of Contracts
  - Using Metrics
- The Contract Design Challenge

Appendix C Design Standard

- The Prime Directive
- Directives
- System Design Guidelines
- Project Design Guidelines
- Project Tracking Guidelines
- Service Contract Design Guidelines

Index