DESIGNING *the* REQUIREMENTS

BUILDING APPLICATIONS THAT THE USER WANTS AND NEEDS

CHRIS BRITTON

Designing the Requirements

Designing the Requirements: Building Applications that the User Wants and Needs

Table of Contents

Contents

Preface

Acknowledgments

About the Author

Chapter 1: Introduction to Context-Driven Design

- Designing Requirements
- What Is Design?
  - Ad Hoc Design
  - Planned Design
  - Engineered Design
  - Summary of Design Approaches
- Making IT Application Development More of an Engineering Discipline
- Taking IT Architecture into Account
- Concluding Remarks

Chapter 2: A Hierarchy of Designs

- Justifying the Hierarchy of Designs
- Context Design
  - Tasks
  - User Groups
  - Data Tables

Pearson

Table of Contents

Messages between Tasks

Task Dependencies

Putting It All Together

Analysis of the Context Design

Integration Design

Technical Design

User Interface Design

Database Design

Implementation

Is It Really Engineering?

Concluding Remarks

Chapter 3: Reusing Existing Methods and Practices

Agile

Individuals and Interactions over Processes and Tools

Working Software over Comprehensive Documentation

Customer Collaboration over Contract Negotiations

Responding to Change over Following a Plan

Conclusion

Upside-Down Design

Use Cases

Atomicity

Confusion of Design Layers

Use Cases Are Confusing

Large Use Case Documents Are Hard to Understand

Use Cases Do Not Assist Engineered Design

Conclusion

The Problem with Estimating Cost

Why Is BDUF Big?

Table of Contents

- Iterations
- Quality
- Testing and Inspection
- Using Existing Practices in Context-Driven Design
- Learning Organizations
- Concluding Remarks

Chapter 4: The Problem of Large Applications

- The Dimensions of Size
- Problems with Large Projects
  - Requirements Problems
  - Lack of End User Support
  - Technical Design Problems
  - Procurement and Outsourcing
- Can Large Projects Be Avoided?
- Concluding Remarks

Chapter 5: The Relationship with the Business

- Understanding Business Processes
- When Its Not a Process
  - Business Services
  - Resource Management
  - Reviewing and Monitoring
- The Need for a Wider View
- Applying the Business Strategy to Application Development
  - Speed of Development
  - Cost versus Performance and Availability
  - Experimental Business Programs
  - How Long Before the Benefits
  - The Need for Security

Table of Contents

Designing for the Existing Culture

Design for a Culture to Which the Organization Aspires

Allow for Changing Plans

Support a Learning Organization

Non-Business Applications

Analysis

Is the Process Well Formed?

Dependency Analysis

Objectives Analysis

Concluding Remarks

Chapter 6: The Relationship with the Users

Adding the Detail

Task Details

Task Fragments

Common Purpose Groups

Data Tables

Messages

Nonfunctional Requirements

Who Uses the Context Design?

Who Are the Users?

Business Process Operations

Monitoring by Management

Data Used by Other Applications

Data Analysis

Application Administration

Analyzing the Context Design

Process Layer

Task Details

Data Table Details

Table of Contents

User Group Details

Message Details

Reviewing the Context Design

Concluding Remarks

Chapter 7: The Relationship to Other IT Projects

Integration Design

Applications

Services

Databases

Services Interface Design

Service Interface Definition

Designing Reusable Services

Existing Applications

Knowing What Is There

Replacing Existing Applications

Fashioning Services from Existing Applications

Looking Back at the Design Process

Concluding Remarks

Chapter 8: User Interface Design and Ease of Use

Logical User Interfaces

From Tasks to Clicks

Ease of Use

Function

Information

Navigation

Text

Help

Intuitive and Likable Applications

Table of Contents

Ease-of-Use Design

Monitoring Ease of Use

Transaction and Task Integrity

The User Interface Design and the Other Detailed Designs

Concluding Remarks

Chapter 9: Database Design

Database Design

Database Design Theory

Programmers versus the Database Designer

Database Access Services

NoSQL

Concluding Remarks

Chapter 10: Technical DesignPrinciples

Principles of High Performance on a Single Machine

Cache

Multithreading and Multiprocessing

Principles of High Performance on Many Servers

Front-End Parallelism

Back-End Parallelism

Principles of High Resiliency

The Need for Testing and Benchmarking

The Technical Design Process

Concluding Remarks

Chapter 11: Technical DesignStructure

Program Structure

What Is a Framework?

The Variety of Programming Languages

Table of Contents

- Choosing a Programming Language and Framework
  - Choose a Language that Fits Your Organizations Skill Set
  - Choose a Language that Is Appropriate for Your Applications Performance Goals
  - Choose a Language that Can Meet Your Integration Requirements
  - Choose a Language that Supports Group Working If Needed
  - Choose Version Control Software and Project Management Software as Well as a Language
  - Choose a Language that Chimes with Your Development Methodology
  - Extending the Framework
  - Implementing Common Functionality
  - Concluding Remarks

Chapter 12: Security Design

- IT Application Security Principles
  - Authentication
  - Access Control
  - User Administration
  - Security Protection
  - Security Monitoring
- The Security Elements of Each Design
  - Context Design
  - Integration Design
  - User Interface Design
  - Database Design
  - Technical Design
- Security Programming
- Concluding Remarks

Chapter 13: The Future of Application Development

- How Context-Driven Design Changes Application Development

Table of Contents

- Context-Driven Design Opportunities
  - New Tools
  - Context and Integration Design
  - User Interface and Database Design
  - Technical Design
- The Application Development Challenges
  - Flexibility
  - Operations
  - Correctness
  - Quality
  - Professionalism
- Concluding Remarks

Appendix A: Context Design Checklist

- Description
  - Elaboration
  - Analysis

References

Index

Pearson