

The Addison-Wesley Signature Series



A VAUGHN VERNON SIGNATURE
BOOK

CONTINUOUS ARCHITECTURE IN PRACTICE

SOFTWARE ARCHITECTURE IN
THE AGE OF AGILITY AND DEVOPS

MURAT ERDER
PIERRE PUREUR
EOIN WOODS

Foreword by KURT BITTNER



Continuous Architecture Principles

Principle 1: Architect products; evolve from projects to products.

Principle 2: Focus on quality attributes, not on functional requirements.

Principle 3: Delay design decisions until they are absolutely necessary.

Principle 4: Architect for change—leverage the “power of small.”

Principle 5: Architect for build, test, deploy, and operate.

Principle 6: Model the organization of your teams after the design of the system you are working on.

Continuous Architecture in Practice: Software Architecture in the Age of Agility and DevOps

Table of Contents

Cover

Half Title

Title

Copyright

Dedication

Contents

Series Editor Foreword

Foreword

Introduction

Acknowledgments

About the Authors

Chapter 1: Why Software Architecture Is More Important than
Ever

What Do We Mean by Architecture?

Software Industry Today

Current Challenges with Software Architecture

Focus on Technology Details Rather than Business Context

Perception of Architects as Not Adding Value

Architectural Practices May Be Too Slow

Some Architects May Be Uncomfortable with Cloud Platforms

Table of Contents

Software Architecture in an (Increasingly) Agile World

The Beginnings: Software Architecture and Extreme Programming

Where We Are: Architecture, Agility, and Continuous Delivery

Where We May Be Going

Introducing Continuous Architecture

Continuous Architecture Definition

Continuous Architecture Benefits

Applying Continuous Architecture

Continuous Architecture Provides a Set of Principles and Tools

Introduction to the Case Study

Case Study Context: Automating Trade Finance

Summary

Chapter 2: Architecture in Practice: Essential Activities

Essential Activities Overview

Architectural Decisions

Making and Governing Architectural Decisions

Quality Attributes

Quality Attributes and Architectural Tactics

Working with Quality Attributes

Building the Quality Attributes Utility Tree

Technical Debt

Capturing Technical Debt

How to Manage Technical Debt

Feedback Loops: Evolving an Architecture

Fitness Functions

Continuous Testing

Common Themes in Today's Software Architecture Practice

Principles as Architecture Guidelines

Table of Contents

- Team-Owned Architecture
- Models and Notations
- Patterns and Styles
- Architecture as a Flow of Decisions

Summary

Chapter 3: Data Architecture

Data as an Architectural Concern

- What Is Data?
- Common Language

Key Technology Trends

- Demise of SQLs Dominance: NoSQL and Polyglot Persistence
- Scale and Availability: Eventual Consistency
- Events versus State: Event Sourcing
- Data Analytics: Wisdom and Knowledge from Information

Additional Architectural Considerations

- Data Ownership and Metadata
- Data Integration
- Data (Schema) Evolution

Summary

Further Reading

Chapter 4: Security as an Architectural Concern

Security in an Architectural Context

- Whats Changed: Todays Threat Landscape
- What Do We Mean by Security?
- Moving Security from No to Yes
- Shifting Security Left

Architecting for Security

- What Is a Security Threat?

Table of Contents

Continuous Threat Modeling and Mitigation

Techniques for Threat Identification

Prioritizing Threats

Other Approaches

Architectural Tactics for Mitigation

Authentication, Authorization, and Auditing

Information Privacy and Integrity

Nonrepudiation

System Availability

Security Monitoring

Secrets Management

Social Engineering Mitigation

Zero Trust Networks

Achieving Security for TFX

Maintaining Security

Secure Implementation

People, Process, Technology

The Weakest Link

Delivering Security Continuously

Being Ready for the Inevitable Failure

Security Theater versus Achieving Security

Summary

Further Reading

Chapter 5: Scalability as an Architectural Concern

Scalability in the Architectural Context

What Changed: The Assumption of Scalability

Forces Affecting Scalability

Types and Misunderstandings of Scalability

The Effect of Cloud Computing

Table of Contents

Architecting for Scalability: Architecture Tactics

- TFX Scalability Requirements

- Database Scalability

- Data Distribution, Replication, and Partitioning

- Caching for Scalability

- Using Asynchronous Communications for Scalability

- Additional Application Architecture Considerations

- Achieving Scalability for TFX

Summary

Further Reading

Chapter 6: Performance as an Architectural Concern

Performance in the Architectural Context

- Forces Affecting Performance

- Architectural Concerns

Architecting for Performance

- Performance Impact of Emerging Trends

- Architecting Applications around Performance Modeling and Testing

- Modern Application Performance Tactics

- Modern Database Performance Tactics

- Achieving Performance for TFX

Summary

Further Reading

Chapter 7: Resilience as an Architectural Concern

Resilience in an Architectural Context

- What Changed: The Inevitability of Failure

- Reliability in the Face of Failure

- The Business Context

- MTTR, Not (Just) MTBF

Table of Contents

MTBF and MTTR versus RPO and RTO

Getting Better over Time

The Resilient Organization

Architecting for Resilience

Allowing for Failure

Measurement and Learning

Architectural Tactics for Resilience

Fault Recognition Tactics

Isolation Tactics

Protection Tactics

Mitigation Tactics

Achieving Resilience for TFX

Maintaining Resilience

Operational Visibility

Testing for Resilience

The Role of DevOps

Detection and Recovery, Prediction and Mitigation

Dealing with Incidents

Disaster Recovery

Summary

Further Reading

Chapter 8: Software Architecture and Emerging Technologies

Using Architecture to Deal with Technical Risk Introduced by New Technologies

Introduction to Artificial Intelligence, Machine Learning, and Deep Learning

Types of Machine Learning

Table of Contents

What about Deep Learning?

Using Machine Learning for TFX

Types of Problems Solved by ML, Prerequisites and Architecture Concerns

Using Document Classification for TFX

Implementing a Chatbot for TFX

Using a Shared Ledger for TFX

Brief Introduction to Shared Ledgers, Blockchain, and Distributed Ledger
Technology

Types of Problems Solved by Shared Ledgers, Prerequisites, and
Architectural Concerns

Shared Ledger Capabilities

Implementing a Shared Ledger for TFX

Benefits of an Architecture-Led Approach

Summary

Further Reading

Chapter 9: Conclusion

What Changed and What Remained the Same

Updating Architecture Practice

Data

Key Quality Attributes

Security

Scalability

Performance

Resilience

The Architect in the Modern Era

Putting Continuous Architecture in Practice

Appendix A: Case Study

Appendix B: Comparison of Technical Implementations of

Table of Contents

Shared Ledgers

Glossary

A

B

C

D

E

F

H

I

K

M

P

Q

R

S

T

U

Index