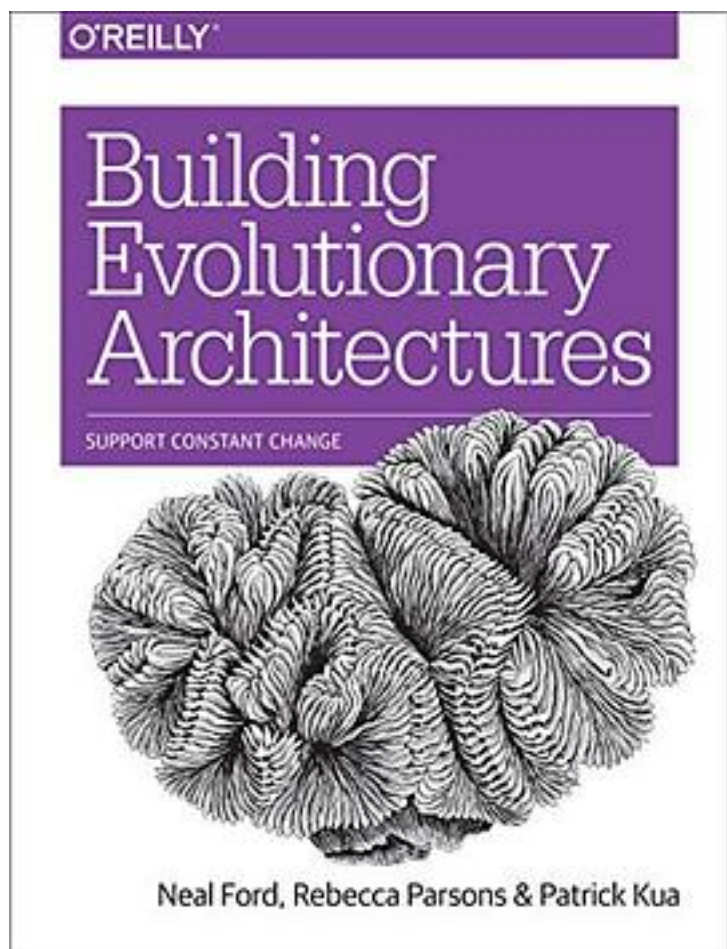


Building Evolutionary Architectures



[Building Evolutionary Architectures_ 下载链接1_](#)

著者:Neal Ford

出版者:O'Reilly Media

出版时间:2017-10-13

装帧:Paperback

isbn:9781491986363

The software development ecosystem is constantly changing, providing a constant stream of new tools, frameworks, techniques, and paradigms. Over the past few years, incremental developments in core engineering practices for software development

have created the foundations for rethinking how architecture changes over time, along with ways to protect important architectural characteristics as it evolves. This practical guide ties those parts together with a new way to think about architecture and time.

作者介绍:

尼尔·福特 (Neal Ford)

是ThoughtWorks软件架构师、Meme Wrangler，曾任DSW集团CTO,是国际公认的软件开发与交付专家。

丽贝卡·帕森斯 (Rebecca Parsons)

是ThoughtWorks CTO，在大规模分布式对象应用开发和系统集成方面拥有丰富经验。

帕特里卡·柯 (Patrick Kua)

是数字银行N26首席科学家，曾任ThoughtWorks主任咨询师和技术主管，在敏捷和精益开发方面拥有丰富经验。

目录: Foreword

Preface

Conventions Used in This Book

O’ Reilly Safari

How to Contact Us

Additional Information

Acknowledgments

1. Software Architecture

Evolutionary Architecture

How Is Long-term Planning Possible When Everything Changes All the Time?

Once I’ ve Built an Architecture, How Can I Prevent It from Gradually Degrading Over Time?

Incremental Change

Guided Change

Multiple Architectural Dimensions

Conway’ s Law

Why Evolutionary?

Summary

2. Fitness Functions

What is a Fitness Function?

Categories

Atomic Versus Holistic

Triggered Versus Continual

Static Versus Dynamic

Automated Versus Manual

Temporal

Intentional Over Emergent

Domain-specific

Identify Fitness Functions Early

Review Fitness Functions

3. Engineering Incremental Change

Building Blocks

Testable
Deployment Pipelines
Combining Fitness Function Categories
Case Study: Architectural Restructuring while Deploying 60 Times/Day
Conflicting Goals
Case Study: Adding Fitness Functions to PenultimateWidgets' Invoicing Service
Hypothesis- and Data-Driven Development
Case Study: What to Port?
4. Architectural Coupling
Modularity
Architectural Quanta and Granularity
Evolvability of Architectural Styles
Big Ball of Mud
Monoliths
Event-Driven Architectures
Service-Oriented Architectures
"Serverless" Architectures
Controlling Quantum Size
Case Study: Guarding Against Component Cycles
5. Evolutionary Data
Evolutionary Database Design
Evolving Schemas
Shared Database Integration
Inappropriate Data Coupling
Two-Phase Commit Transactions
Age and Quality of Data
Case Study: Evolving PenultimateWidgets' Routing
6. Building Evolvable Architectures
Mechanics
1. Identify Dimensions Affected by Evolution
2. Define Fitness Function(s) for Each Dimension
3. Use Deployment Pipelines to Automate Fitness Functions
Greenfield Projects
Retrofitting Existing Architectures
Appropriate Coupling and Cohesion
Engineering Practices
Fitness Functions
COTS Implications
Migrating Architectures
Migration Steps
Evolving Module Interactions
Guidelines for Building Evolutionary Architectures
Remove Needless Variability
Make Decisions Reversible
Prefer Evolvable over Predictable
Build Anticorruption Layers
Case Study: Service Templates
Build Sacrificial Architectures
Mitigate External Change
Updating Libraries Versus Frameworks
Prefer Continuous Delivery to Snapshots
Version Services Internally
Case Study: Evolving PenultimateWidgets' Ratings
7. Evolutionary Architecture Pitfalls and Antipatterns

Technical Architecture
Antipattern: Vendor King
Pitfall: Leaky Abstractions
Antipattern: Last 10% Trap
Antipattern: Code Reuse Abuse
Case Study: Reuse at PenultimateWidgets
Pitfall: Resume-Driven Development
Incremental Change
Antipattern: Inappropriate Governance
Case Study: Goldilocks Governance at PenultimateWidgets
Pitfall: Lack of Speed to Release
Business Concerns
Pitfall: Product Customization
Antipattern: Reporting
Pitfall: Planning Horizons
8. Putting Evolutionary Architecture into Practice
Organizational Factors
Cross-Functional Teams
Organized Around Business Capabilities
Product over Project
Dealing with External Change
Connections Between Team Members
Team Coupling Characteristics
Culture
Culture of Experimentation
CFO and Budgeting
Building Enterprise Fitness Functions
Case Study: PenultimateWidgets as a Platform
Where Do You Start?
Low-Hanging Fruit
Highest-Value
Testing
Infrastructure
Case Study: Enterprise Architecture at PenultimateWidgets
Future State?
Fitness Functions Using AI
Generative Testing
Why (or Why Not)?
Why Should a Company Decide to Build an Evolutionary Architecture?
Case Study: Selective Scale at PenultimateWidgets
Why Would a Company Choose Not to Build an Evolutionary Architecture?
Convincing Others
Case Study: Consulting Judo
The Business Case
“The Future Is Already Here...”
Moving Fast Without Breaking Things
Less Risk
New Capabilities
Building Evolutionary Architectures
Index
• • • • • ([收起](#))

[Building Evolutionary Architectures_ 下载链接1](#)

标签

架构

软件工程

软件开发

architecture

计算机

Software_Architecture

编程

ThoughtWorks

评论

虽然是我司出品……但是……还是一般……

这书挺适合大规模系统架构的architect新手的，提供了十分有益的high-level视角，日后不至于被各种细节淹没，跳不出来。当然，已经有了丰富经验的人读它，就会觉得太浅了。

Day 12 干货比较少，只学到了一个新概念，fitness function #百日早起学习挑战

#平衡之道

忽悠的多了，就空洞；Rebecca奶奶也胡来了

evolutionary = guided + incremental change。实际就是在自动化发布集成的基础上增加各个方面的衡量指标。道理很简单，也没有讲很多practical的例子。更多的是大面积的科普，可作为深入阅读的入口。

值得再读下中文版

挺不错的，既讲了架构，也讲了组织、业务和实践，虽然每个点都没有特别深入，但整体讲的很全面又很有条理，感觉把之前看的很多内容都串起来了。核心的观点就是把演化性作为系统架构设计的一个维度来思考，通过fitness function只指引系统不断往好的方向演化

把软件架构方方面面的东西都讲到了，有开发经验的都能从书中找到吻合的地方

只能讲还行。Fitness function用来衡量和评价架构决策，是提出的个不错的概念。

[Building Evolutionary Architectures 下载链接1](#)

书评

《Building Evolutionary Architectures》这本书大概翻译过来是《设计可进化架构》。这本书虽然目标读者是系统架构师，但是也推荐从业三五年的工程师阅读。非常不推荐在校大学生或者刚刚毕业的工程师看这本书，因为这本书实例很少，只有做过很多项目、也在很多项目上摔过跤的人...

整本书其实就是一个大的idea -
变化无法避免，让我们把适应变化作为架构设计的一个原生维度来考虑 -
这个写一篇文章即可 - 写一本书实在是。。。英文版就很啰嗦，翻译的版本就更难读了
- 两星给英文版，一星给中文版。字数补丁 字数补丁 字数补丁 字数补丁 字数补丁
字数补丁...

[Building Evolutionary Architectures_下载链接1_](#)