

CONTENTS IN DETAIL

Introduction	xv
1 Time Is a Flat Circle	1
Leveraging What Came Before	3
The User as the Market Co-Creator	8
The Mainframe and the Cloud	9
The Flat-Rate Internet	11
Migrating for Value, Not for Trends	14
2 Cannibal Code	17
Alignable Differences and User Interfaces	18
Unix Eats the World	21
Inheritance Paths	27
Leveraging Interfaces When Approaching Legacy Systems	32
Beware Artificial Consistency	35
3 Evaluating Your Architecture	37
Problem 1: Technical Debt	38
Example: The General Ledger	40
Problem 2: Performance Issues	42
Example: Case Flow Management	44
Problem 3: Stability Issues	45
Example: Custom Configuration	49

Stages of a Modernization Plan	51
No Silver Bullets	53
Full Rewrite	54
Iteration in Place.	55
Split in Place.	56
Blue-Green	56
The Hard Cutoff.	57
Putting It Together.	57
4 Why Is It Hard?	59
The Curse of Hindsight.	60
Easy and Also Impossible	63
Overgrowth: The Application and Its Dependencies	64
Shifting Vertically: Moving from One Abstraction Layer to Another	66
Shifting Horizontally: Moving from One Application to Another	67
Shifting from Client to Server	67
Shifting Down the Dependency Tree.	68
Cutting Back the Overgrowth.	68
Automation and Conversion	69
Transpiling Code	69
Static Analysis.	71
A Guide to Not Making Things Harder.	72
5 Building and Protecting Momentum	75
Momentum Builder: The Bliss of Measurable Problems.	76
Anatomy of the Measurable Problem	78
Momentum Killer: The Team Cannot Agree	80
Step 1: Define a Scope	81
Step 2: Check for Conflicting Optimization Strategies	83
Step 3: Perform Time-Boxed Experiments	83

Momentum Killer: A History of Failure	84
Momentum Builder: Inspiring Urgency	87
Protecting Momentum: A Quota on Big Decisions	89
Protecting Momentum: Calculating Opportunity Costs	90
The Cost of Not Adding New Features	93
The Cost of Not Fixing Something Else	94
The Cost of Not Deprecating in Favor of a Different Solution	95
6 Coming in Midstream	97
Finding the Bleed	98
Mess: Fixing Things That Are Not Broken	101
Figuring Out Whether Something Needs to Be Fixed.	105
But . . . What About Conventions?.	106
When Does Breaking Up Add Value?.	107
The Compounding Problem: Diminishing Trust	108
Solution: Formal Methods.	109
Mess: Forgotten and Lost Systems	111
The Compounding Problem: Crippling Risk Avoidance	112
Solution: Chaos Testing	112
Mess: Institutional Failures	114
The Compounding Problem: No Owners	115
Solution: Code Yellow.	116
Calling a Code Yellow.	120
Running a Code Yellow.	121
Mess: Leadership Has Lost the Room	122
The Compounding Problem: Self-Sabotaging Teams	125
Solution: Murder Boards.	125
Stopping the Bleed.	127
7 Design as Destiny	129
Designing Technical Conversations	131
How to Run a Design Exercise	132

More About Follow-ups: Why vs. How	134
Some Useful Design Exercises for Engineering Teams	134
Exercise: Critical Factors	135
Exercise: The Saboteur	135
Exercise: Shared Uncertainties	136
Exercise: The 15 Percent	137
Exercises Specifically for Decisions	138
Exercise: Probabilistic Outcome-Based Decision Making	138
Exercise: Affinity Mapping	139
Team Structure, Organization Structure, and Incentives	140
Individual Incentives	142
Minor Adjustments as Uncertainty	145
Organization Size and Communication	147
Manager Incentives	148
Designing a Team: Applications of Conway's Law	149
Reorgs Are Traumatic	151
Finding the Right Leadership	152
Exercise: The Smallest Testable Unit	153
Structuring the Team to Account for Past Failure	155
Exercise: In-Group/Out-Group	158
Takeaways	159
8 Breaking Changes	161
Being Seen	162
Who Draws the Line?	167
Building Trust Through Failure	168
Breaking Change vs. Breaking	170
Why Break Things on Purpose?	171
Projecting Impact	172
The Kill Switch	177
Communicating Failure	178
Failure Is a Best Practice	179

9 How to Finish	181
Revealing Assumptions	182
Approach 1: Success Criteria	182
Example: Adding Continuous Integration/ Continuous Deploy	183
Approach 2: Diagnosis-Policy-Actions.	184
Example: Upgrading a Database.	184
Comparison	185
Marking Time	186
Postmortems on Success.	187
Postmortem vs. Retrospective	188
Running Postmortems	188
The Tale of Two War Rooms.	190
Working Groups vs. Committees	191
Success Is Not Obvious If Undefined	194
10 Future-Proofing	195
Time	196
Unescapable Migrations.	199
Failing Gracefully	200
Less Time Between Upgrades, Not More	202
A Word of Caution About Automation.	206
Building Something Wrong the Correct Way	207
Feedback Loops.	210
Don't Stop the Bus.	214
Conclusion	217