

---

# The Journey to Enterprise Agility

---

Daryl Kulak • Hong Li

# The Journey to Enterprise Agility

Systems Thinking  
and Organizational Legacy

Daryl Kulak  
Westerville, OH  
USA

Hong Li  
Westerville, OH  
USA

ISBN 978-3-319-54086-3      ISBN 978-3-319-54087-0 (eBook)  
DOI 10.1007/978-3-319-54087-0

Library of Congress Control Number: 2017937541

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer International Publishing AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*We dedicate this book to Robert Rosen and W. Edwards Deming for their contributions to systems science, the workplace and society. Each man forged a new path, even though they were often marginalized and misinterpreted during their lives and even afterward. Each of them saw derivative works that claimed to represent them, but often fell far short. We hope to guide the conversation back to Rosen's and Deming's original material, where the real value still lies.*

---

## Preface

I remember Hong's frantic phone call back in 2003. I was busy trying to get a new solo consulting venture off the ground. Hong wanted to talk about some "systems thinking" nonsense. He wanted us to write a damn book about it. I had just finished the second edition of my first book and, quite frankly, was in no mood.

Three years later, the ever-persistent Hong had convinced me this was a book that needed to be written.

Hong warned me this would be a long journey. I had much to learn, because I had to catch up on his 3 years of (full-time) research. We then agreed to meet every Sunday morning at around 10:30. In 2006, we started writing in earnest on evenings and weekends.

So, we began in 2006. Eleven years ago, as I write these words.

It took me a long, long time to understand systems thinking as Hong was teaching me. It took even longer to apply it to day-to-day life in a software team. I am a slow learner, apparently. Glacial. But, eventually, it came together in my head. I began to feel like we had something big.

Around 2008, Hong and I trashed what we had and rewrote the entire book once we realized that our audience might not want to know every thought, meal and fart of our systems scholars. We rewrote it again the next year to incorporate more storytelling into the text, rather than dry bullet points of principles and concepts. We figured out better, clearer ways to state our message. Another rewrite. And another. The versions are there in Google Docs if you want to see them (it was called Writely back then).

Every year, I would think that next year would be the year. We would finish it, damn it. I would even update my LinkedIn status to show that my next book would be released in 2011, I mean, 2012, I mean 2013. . . This was getting embarrassing. I started dreading that question at parties, "So, how's the book going?"

But now it's 2017. We've finished it. So there's a fair bit of work sitting here in front of you. We've talked about these ideas with all our clients over these 11 years, and they've helped us refine them. We've had dialogs over lunch and drinks with our colleagues and fellow travelers. We've done presentations at conferences. Fortunately, we joined a consulting company, Pillar Technology, who encouraged everything we were doing and actually incorporated a lot of our ideas into the company's way of managing projects. And we've used the ideas day-to-day in our consulting work to continually improve the execution of every project we've led

over these years. Many other friends and coworkers have helped us understand how this book needed to be written.

We won't say that it works. We can't say that. It isn't a set of best practices. But we hope we can help provide a new worldview and yes, some practices that could be helpful to our readers as you try to push some software out the door.

Westerville, OH  
January 2017

Daryl Kulak

---

## Acknowledgments

We've been working on the ideas in this book for a long time and, thus, have had many collaborators. We will inevitably miss some names and, for that, we are sorry.

Thank you so much to Chris Davis, Geoff Wilhelm, Yvonne Weller, Mike Miller, Naum Sayfullin, Conal Thompson, Joe Sjostrom, Diane Roquemore, Jodi Cannon, Rob Richardson, Zach Guisinger, Andy Secrest, Chris Beale, Patrick Welsh, Gary Baker, Justin Searls, Kelly Allan, Sanjiv Augustine, Rick Neighborger, Angelo Mazzocco, Ben Blanquera, Kermit Morse, Justin Foley, John Griffin, Jim Hertzfeld, Anita Shankar, Rich Diers, Bill Gray, Sip Reyes, Rachel Howard, Mike Cottmeyer, Mark Davidson, Mark Walker, Joe Hammond and Tax Commissioner Joe Testa.

Thank you to the readers of Daryl's first book and everyone who gave us encouragement to write another one.

Thanks to our colleagues at ISSS, who continually encourage us to progress in our understanding of systems thinking, including Dennis Finlayson, Delia MacNamara, Debora Hammond, Jennifer Makar and many others.

A special thanks to Linda Farrenkopf, Brian Caldwell, Walter den Haan, Jeff Jamison and John Lavkulich for your enduring friendship and encouragement.

Thanks to everyone at Pillar for allowing us to try our ideas with so many clients, incorporating our work into the very fabric of how Pillar operates. We appreciate the conversations and feedback from many Pillarites, including Bob Myers, Matt VanVleet, Gary Gentry, John Huston, Katie Robinson, Allen Smith, Cheryl Smith, Nish Gandhi, Mary Kaufman, Dan Wiebe, Todd Flanders, Kevin Smith, Steve Yaffe, Rich Dammkoehler and Kevin McCann.

Thank you to everyone in the Java Guys program, organizers and inmates, at Marion Correctional Institute. You guys really inspire us with your continuous dedication to becoming great test-driven developers.

Thanks in particular to Gene Johnson for joining "The Church of Hong" those Sunday mornings and giving us specific, but always kind, feedback on so many parts of this book. You really helped us organize our ideas.

We'd like to thank Dr. Theodore Williams for creating the Purdue Enterprise Reference Architecture (PERA), for introducing Hong to the major players in enterprise integration, and for being a true gentleman in everything he did.

Thank you to Jason Kinsey for providing the creative illustrations for this book. Jason, you did a great job on a short time schedule. You are so very talented.

A big, amazing thank you to Springer-Verlag and especially editor Ralf Gerstner and his team. Thank you for giving us this chance to take our ideas to a worldwide audience under your guidance and leadership.

And thanks, of course, to our wives and families for supporting us and loving us through the trials and tribulations of writing this book.



---

## Endorsements

A masterful, thought-provoking read for any business leader motivated to sustainably improve business value. Its delightfully entertaining but substance-rich style will readily guide you to “aha moments” on how to advance agile software development to enterprise scale!

**Conal Thompson**, CTO and VP of IT, Chemical Abstracts Service,  
a Division of the American Chemical Society

At recent conferences of the International Society for the Systems Sciences (ISSS), I have enjoyed talking with Daryl, learning about his work, and witnessing the evolution of this book. I’m impressed with the way he and co-author Hong have effectively linked abstract concepts of systems theories with good, practical advice for software development teams. They’ve created a valuable bridge between academic scholarship in the systems field and the day-to-day functioning of software teams. A must-read for software managers.

**Debora Hammond**, PhD, Professor of Interdisciplinary Studies,  
Hutchins School of Liberal Studies, Sonoma State University;  
author of “The Science of Synthesis” and past president of ISSS

Dr. Li and Daryl have produced a very nice book in “The Journey to Enterprise Agility.” This book integrates the highest priority ingredients that are necessary to achieve true enterprise agility. My favorite ingredient is human relationships. This book not only includes important teaching on process, technology, value, etc. but also on human relationships. It is no wonder that Hong and Daryl have not only written this excellent book but also that they have spent their careers successfully developing systems within an enterprise agility framework.

**Angelo Mazzocco**, CIO, Central Ohio Primary Care

This is a marvelous book! It only took a few pages to realize that the authors have created a sound basis for effective team performance. The Journey to Enterprise Agility provides a mindset, not a static formula, which is refreshing. Any software development organization that chooses not to embrace what this book offers probably misses an opportunity to enhance their performance.

**Gene Bellinger**, Director - Systems Thinking World ([systemswiki.org](http://systemswiki.org))

Have you wondered how to improve the process of software product development? It's about changing your approach to leadership, your concept of professionalism, even how you see the world. But it's worth it. Daryl and Hong's book will give you details on how to make that shift.

**Philip Pointer**, VP Applications Development, Mednax National Medical Group

---

# Contents

<b>1</b>	<b>Today's Problems with Enterprise Business Software . . . . .</b>	<b>1</b>
1.1	The New, New Thing . . . . .	2
1.2	Cheap Green Shirts . . . . .	3
1.3	Help! We're Terrible! . . . . .	4
1.4	Aristotle, Descartes and Disconnection to Business Value . . . .	5
1.5	The Mechanical Business World . . . . .	8
1.5.1	The Question of Business Value . . . . .	9
1.6	Scalability and Sustainability . . . . .	11
1.6.1	The Story of Sticky LaGrange . . . . .	11
1.7	Yes, But What About the Illth? . . . . .	13
1.7.1	Agile Illth . . . . .	13
1.8	Our Software Industry Problems Can Be Overcome . . . . .	15
	Reference . . . . .	16
<b>2</b>	<b>The Scholars of Systems Thinking . . . . .</b>	<b>17</b>
2.1	Hard and Soft Systems Thinking . . . . .	17
2.1.1	Don't Worry: This Will Not Be a Complete History of Systems Thinking . . . . .	21
2.2	Systems Thinking Forms the Basis . . . . .	22
<b>3</b>	<b>Worldview and Intentions . . . . .</b>	<b>23</b>
3.1	Borrowing from the Buddha . . . . .	23
3.1.1	Right View . . . . .	24
3.1.2	Right Intention . . . . .	28
3.1.3	Right Speech . . . . .	29
3.1.4	Right Action . . . . .	31
3.2	Right View + Right Intention + Right Speech + Right Action . . .	32
3.2.1	A Worldview That Is Compatible with Success . . . .	32
	Reference . . . . .	33
<b>4</b>	<b>Seven Principles of Systems Thinking for Software Development . . .</b>	<b>35</b>
4.1	So Many Principles! . . . . .	35
4.1.1	Systems Thinking Principle 1: Trust = Speed . . . .	36
4.1.2	Systems Thinking Principle 2: Avoid Best Practices . . . . .	37

4.1.3	Systems Thinking Principle 3: Beware the Immense Power of Analogies . . . . .	39
4.1.4	Systems Thinking Principle 4: Blame the System, Not the Person . . . . .	40
4.1.5	Systems Thinking Principle 5: Treat People Like People, Not Like Machines . . . . .	41
4.1.6	Systems Thinking Principle 6: Acknowledge Your Boundaries . . . . .	42
4.1.7	Systems Thinking Principle 7: Relation-ness Matters More Than Thing-ness . . . . .	43
4.2	Principles for Your Worldview . . . . .	44
	References . . . . .	45
<b>5</b>	<b>Redefining Professionalism . . . . .</b>	<b>47</b>
5.1	Understanding What It Means to Be a Professional . . . . .	47
5.1.1	What Defined the Professionalism of the Past? . . . .	48
5.1.2	Mechanical Professionalism . . . . .	49
5.2	The New Professionalism . . . . .	52
5.3	The Principles of the New Professionalism . . . . .	53
5.3.1	New Professionalism Principle 1: Speak Up! . . . . .	54
5.3.2	New Professionalism Principle 2: Solving Communication Problems with Via Negativa . . . . .	59
5.3.3	New Professionalism Principle 3: Be an Advocate for Weak or Absent Voices . . . . .	63
5.3.4	New Professionalism Principle 4: Proudly Display Your Dirty Laundry . . . . .	70
5.3.5	New Professionalism Principle 5: Connect People to One Another . . . . .	70
5.3.6	New Professionalism Principle 6: Challenge Your Own Assumptions as Much as You Challenge Others' . . . . .	71
5.3.7	New Professionalism Principle 7: Be Accountable to Change . . . . .	72
5.3.8	New Professionalism Principle 8: Manage Uncertainties Through Adaptive Practices and Stop Faking Risk Management . . . . .	73
5.3.9	The Worldview of the New Professional . . . . .	78
	References . . . . .	79
<b>6</b>	<b>Scaling and Sustaining: Avoiding Mechanical Behavior . . . . .</b>	<b>81</b>
6.1	The Burning Question on Robert Rosen's Mind . . . . .	82
6.2	Allow Me to Work Somewhere Fit for Humans . . . . .	83
6.3	Are Humans Similar to Software? . . . . .	83
6.4	But Agile Ain't Mechanical, Is It? . . . . .	85
6.5	Conversations with a Terrible Coach . . . . .	85
6.6	Why? What's the Purpose? . . . . .	87

6.6.1	Thin Knowledge Versus Thick Knowledge . . . . .	87
6.6.2	Why? Tell Me Your Thought Process . . . . .	88
6.7	What Type of Organization Do You Work In? . . . . .	89
6.8	Agile Practices and the “Why” Behind Them . . . . .	90
6.8.1	Be a Skeptical Empiricist . . . . .	91
6.8.2	Run a Process Experimentation Lab in Every Team Space . . . . .	92
	References . . . . .	93
<b>7</b>	<b>Business Value, Estimation and Metrics . . . . .</b>	<b>95</b>
7.1	Do We Really, Honestly Need a PMO? . . . . .	95
7.2	How an Idea Becomes a Project . . . . .	96
7.3	The Problems with Today’s Portfolio Management Processes Are . . . . .	96
7.3.1	The Annual Portfolio Management Cycle . . . . .	96
7.3.2	Projects Incur Unnecessary Costs and Risks . . . . .	97
7.3.3	Early Estimates Are Inaccurate . . . . .	97
7.4	Ideas for Portfolio Management . . . . .	98
7.4.1	The Fleeting Concept of Value . . . . .	98
7.4.2	Value Stories . . . . .	99
7.4.3	The Cisco Rule . . . . .	102
7.4.4	Prioritizing Value Stories . . . . .	103
7.4.5	Slicing Value Stories . . . . .	103
7.4.6	The Real Day-to-Day Magic of Value Stories . . . . .	104
7.4.7	“Get Me a Black Truck!” . . . . .	106
7.4.8	Three Levels of ROI: Thin, Thick and “Thurmanator” . . . . .	107
7.4.9	Metrics . . . . .	109
7.4.10	Estimation . . . . .	117
7.5	Meanwhile. . . Back in the Team Space . . . . .	121
7.6	Portfolio Management Does Not Have to Be a Dinosaur . . . . .	122
	References . . . . .	122
<b>8</b>	<b>Missing Deadlines Means Missing Market Opportunities . . . . .</b>	<b>123</b>
8.1	How to Miss a Deadline . . . . .	123
8.2	Why Requirements Is a Bad Word . . . . .	124
8.2.1	Shotgun or Rifle Approach? . . . . .	126
8.2.2	Adding a Sponge to the Iron Triangle . . . . .	126
8.2.3	Acceptable and Unacceptable Responses to the Business . . . . .	128
8.2.4	The Steel Query Application . . . . .	128
8.3	“How Much Will It Cost?” Is the Wrong Question . . . . .	130
8.3.1	Thinking in Buckets . . . . .	131
8.3.2	Short Sprints Really Help . . . . .	133

8.4	Stop Thinking Like an IT Person: Think Like a Businessperson . . . . .	133
8.5	Things Move So Dang Fast . . . . .	134
	Reference . . . . .	135
<b>9</b>	<b>Flipping the Run/Build Ratio: The Business Case for Software Craftsmanship . . . . .</b>	<b>137</b>
9.1	Just Keeping the Lights On . . . . .	137
9.2	Software Craftsmanship: How a Movement Among Developers Is Good for Business . . . . .	139
9.2.1	Software Craftsmanship Practices . . . . .	141
9.2.2	Software Craftsmanship with Commercial Packaged Software . . . . .	147
9.3	DevOps . . . . .	149
9.4	The Intention of Software Craftsmanship . . . . .	150
	References . . . . .	151
<b>10</b>	<b>Better Vendor RFPs and Contracts . . . . .</b>	<b>153</b>
10.1	Let's Put It Out to Bid! . . . . .	153
10.2	The Alan Shepard Principle . . . . .	154
10.3	A Better Way to Write RFPs . . . . .	155
10.3.1	RFPs with Value Stories . . . . .	155
10.3.2	Sprint Zero Contract with Follow-On . . . . .	157
10.3.3	Connecting Non-Agile Vendors to Your Agile Process . . . . .	157
10.4	Working with Vendors Is Not Always Easy . . . . .	161
<b>11</b>	<b>Servant Leadership . . . . .</b>	<b>163</b>
11.1	The Basis of Servant Leadership . . . . .	163
11.1.1	Needs Not Wants . . . . .	165
11.1.2	Management and Leadership . . . . .	166
11.2	A Leadership Culture . . . . .	167
11.3	The Nine Leadership Principles . . . . .	168
11.3.1	Leadership Principle 1: Bring Decision-Making as Close to the Work as Possible . . . . .	168
11.3.2	Leadership Principle 2: A Leader Is Anxious When the Team Is Calm and Calm When the Team Is Anxious . . . . .	171
11.3.3	Leadership Principle 3: A Leader Is a Systems Scientist . . . . .	171
11.3.4	Leadership Principle 4: A Leader Understands the Benefits and Dangers of Measurement . . . . .	172
11.3.5	Leadership Principle 5: A Leader Negotiates . . . . .	174
11.3.6	Leadership Principle 6: A Leader Leads Across Team Boundaries Using Scouts and Ambassadors . . . . .	175

11.3.7	Leadership Principle 7: A Leader Organizes Field Trips . . . . .	177
11.3.8	Leadership Principle 8: A Leader Is a Student of Corporate Culture . . . . .	178
11.3.9	Leadership Principle 9: A Leader Respects the Psychological Contract . . . . .	179
11.4	The Desolate Wasteland of Management Training . . . . .	180
11.5	Getting Started with Servant Leadership . . . . .	181
	References . . . . .	182
<b>12</b>	<b>How Teams Keep Learning and Improving . . . . .</b>	<b>183</b>
12.1	Continuous Improvement with W. Edwards Deming . . . . .	183
12.1.1	The Shewhart Cycle . . . . .	184
12.1.2	The Trouble with Self-Reinforcing Feedback . . . . .	192
12.2	Where's the System? . . . . .	193
12.3	How Long Is a Sprint? . . . . .	193
12.4	Lightning Talks . . . . .	195
12.5	Learning, Improving . . . . .	195
	References . . . . .	196
<b>13</b>	<b>Getting Coaching That Really Helps . . . . .</b>	<b>197</b>
13.1	Fighting Poverty in Rural China . . . . .	197
13.2	How to Transform an Organization? . . . . .	199
13.2.1	Shu-ha-ri . . . . .	199
13.2.2	The Alternative to Shu-ha-ri . . . . .	200
13.2.3	Scrum and Done! . . . . .	201
13.2.4	Aspects of a Transformation . . . . .	202
13.3	Coaches Who Just Coach: NO; Extensive Training Sessions: NO; Certification: HELL NO . . . . .	203
13.4	Talker+Doer . . . . .	205
13.5	Six Questions to Ask Your Agile Coach or Consultant . . . . .	205
13.6	People Don't Resist Change . . . . .	208
	Reference . . . . .	209
<b>14</b>	<b>Capitalizing Software Investments . . . . .</b>	<b>211</b>
14.1	Will That Be CAPEX or OPEX? . . . . .	211
14.1.1	CAPEX and OPEX Activities in Software Development . . . . .	212
14.2	Why Capitalize? . . . . .	213
14.2.1	Income Smoothing . . . . .	213
14.2.2	Ratio Analysis . . . . .	214
14.2.3	Reduced OPEX . . . . .	214
14.3	Creating Financial Projections with More Confidence . . . . .	214

<b>15</b>	<b>Integrating Enterprise Methodology and Architecture with Fast-Moving Development Teams . . . . .</b>	<b>217</b>
15.1	The Organizational Legacy . . . . .	217
15.1.1	Playing Nice: Avoiding Process Illth . . . . .	218
15.2	How Development Teams Can Play Nice with Corporate Methodology Groups . . . . .	218
15.2.1	Stage Gates . . . . .	219
15.2.2	Project Plans . . . . .	221
15.2.3	Corporate Methodology Is Not the Enemy . . . . .	225
15.3	How the Corporate Methodology Group Can Transform Its Worldview . . . . .	226
15.4	How Development Teams Can Play Nice with Enterprise Architecture . . . . .	227
15.4.1	Architects: Come to Our Demos! . . . . .	227
15.4.2	Volunteer to Be an Early Adopter for New Architectural Capabilities . . . . .	228
15.5	How the Enterprise Architecture Group Can Transform Its Worldview . . . . .	228
15.5.1	Architects Who Write Code . . . . .	228
15.5.2	Setting Standards . . . . .	229
15.5.3	Emergent Architecture . . . . .	232
15.5.4	What's That Sound? Is Your Architecture Screaming? . . . . .	232
15.5.5	Test-Driven Architecture . . . . .	233
15.5.6	Being an Architect Is Not the Desired Career Path for Every Developer . . . . .	233
15.6	A Word About Bimodal IT . . . . .	234
15.7	The Corporate Immune System . . . . .	234
	Reference . . . . .	235
<b>16</b>	<b>HR Agility . . . . .</b>	<b>237</b>
16.1	Why Did Churchman Quit the Field He Pioneered? . . . . .	238
16.2	Changing Your HR Worldview: People Aren't Your Resources . . . . .	239
16.2.1	The Neighborhood Nurses . . . . .	240
16.3	Evaluating People . . . . .	241
16.3.1	Reduce the Document Focus . . . . .	241
16.3.2	Performance Appraisals . . . . .	242
16.4	Rewarding People . . . . .	244
16.4.1	Goal-Setting and Specific Targets . . . . .	244
16.4.2	Tangible Rewards . . . . .	245
16.4.3	Work That Is Challenging, but Not Overwhelming . . . . .	247
16.4.4	The Fun Committee . . . . .	249
16.5	Teams Competing Against One Another . . . . .	249
16.6	Promotion Paths . . . . .	250
16.6.1	Leadership Promotion Paths . . . . .	251



16.7	There Is Only One Motivator . . . . .	251
16.8	Recruiting, Interviewing and Contracting . . . . .	252
16.8.1	Diversity and Inclusion . . . . .	252
16.8.2	Don't Fetishize Technical Skills in Interviews . . . . .	253
16.8.3	The Use of Contractors and Contracted Services . . . . .	255
16.9	Changing the Culture . . . . .	255
16.9.1	Look! I'll Change the Culture by Replacing the People! . . . . .	256
16.9.2	Being a Student of Corporate Culture . . . . .	256
16.10	How Team Roles Change . . . . .	257
16.10.1	Thinner Management Layers . . . . .	257
16.10.2	User Interface Designers . . . . .	258
16.10.3	Generalizing Specialists . . . . .	259
16.11	Allocating People's Time . . . . .	260
16.11.1	Competency Centers . . . . .	260
16.11.2	Time-Slicing People Is the Biggest Productivity Killer in the Known World . . . . .	261
16.12	HR Is a Big Part of an Agile Transformation . . . . .	263
	References . . . . .	264
<b>17</b>	<b>Buy Versus Build . . . . .</b>	<b>265</b>
17.1	The Character of Your IT Shop . . . . .	265
17.2	To Buy or to Build? . . . . .	266
17.2.1	What Business Are You In? . . . . .	267
17.3	It Might Not Be Buy Versus Build: It Could Be Buy and Build . . . . .	270
17.4	Where Are You on the Emotional Spectrum of Packaged Software? . . . . .	271
	Reference . . . . .	272
<b>18</b>	<b>A Brief Note About Using Offshore Teams . . . . .</b>	<b>273</b>
18.1	Mind-Body Separation . . . . .	273
18.1.1	Slice at the Thinnest Part . . . . .	274
18.2	High and Low Bandwidth Communication . . . . .	276
18.2.1	Those Tricky Metaphors . . . . .	277
18.3	Large Distributed Teams Can Work . . . . .	278
<b>19</b>	<b>Highlighting the Differences Between Software Product Companies and Internal IT . . . . .</b>	<b>279</b>
19.1	The Role of Software in Your Organization . . . . .	279
19.1.1	Making Money by Selling Software Versus a Software Cost Center . . . . .	280
19.1.2	A Software Subsidiary . . . . .	282
19.2	Product Managers and Product Owners . . . . .	282
19.3	Selling Software or Just Using It . . . . .	283
<b>20</b>	<b>Conclusion . . . . .</b>	<b>285</b>

The Journey to Enterprise Agility  
Systems Thinking and Organizational Legacy

Kulak, D.; Li, H.

2017, XIX, 286 p. 35 illus., 32 illus. in color., Hardcover

ISBN: 978-3-319-54086-3