



THEORY IN PRACTICE

Beautiful Teams

团队之美（影印版）

资深团队领导者讲述发人深省和引以为戒的故事

O'REILLY®

东南大学出版社

Andrew Stellman & Jennifer Greene 编

团队之美 (影印版)
Beautiful Teams



Jennifer Greene

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

东南大学出版社

图书在版编目 (CIP) 数据

团队之美: 英文 / (美) 斯特尔曼 (Stellman, A.),
(美) 格林尼 (Greene, J.) 著. —影印本. —南京: 东南
大学出版社, 2010.6

书名原文: Beautiful Teams

ISBN 978-7-5641-2270-6

I. ①团… II. ①斯… ②格… III. ①软件工程—组
织管理—英文 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 089046 号

江苏省版权局著作权合同登记

图字: 10-2010-155 号

©2009 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2010. Authorized reprint of the original English edition, 2009 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2009。

英文影印版由东南大学出版社出版 2010。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制或。

团队之美 (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江 汉

网 址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 31.75 印张

字 数: 786 千字

版 次: 2010 年 6 月第 1 版

印 次: 2010 年 6 月第 1 次印刷

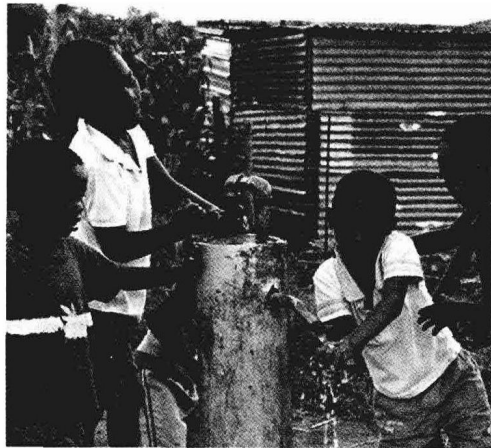
书 号: ISBN 978-7-5641-2270-6

印 数: 1~1800 册

定 价: 64.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

This book is dedicated to Trevor Field and the rest of the people at PlayPumps International for all of their dedication and hard work bringing clean drinking water to the people of sub-Saharan Africa. A portion of every sale of Beautiful Teams will be donated to their worthy cause. We urge you to learn more about them and their work by visiting <http://www.playpumps.org>.



Why Beautiful Teams?

WE’VE BEEN ON A LOT OF DIFFERENT TEAMS OVER THE YEARS, IN A LOT OF DIFFERENT COMPANIES AND building a lot of different kinds of projects. And over the course of writing books, articles, and blog posts about how to make projects run better, we were always nagged by a question. It always seemed like it was our job to come up with prescriptive “best” ways to run software projects: how to plan the projects, how to build the software, and how to make sure that it doesn’t break. But the more we wrote and the more we talked to people, the more we questioned this idea.

We started down that path after writing our first book, which we thought of as a practical recipe book for running successful software projects. We’d done a lot of research, experimentation, and real-world project work over the years to find practices that worked for us: ways to plan software projects, techniques for developers to write better code, and ways to test the software. We took the ones that worked best for us and packaged them up into as lightweight a book as we could come up with. We’ve gotten a lot of great, overwhelmingly positive feedback about it over the years, and a lot of people have told us that they use it every day.

And that’s where things started to go wrong for us.

Ironically, we fell into a trap that we actually wrote about in that book: we started to get a kind of tunnel vision. We saw this particular set of practices as the “successful” practices. We never intended to say that there was only one way to plan out your project, or to estimate it, or to do a code review, or to test the software. But what we found, as we started getting out into the world and talking to more and more software people, is that we were getting pigeonholed. People would say, “Oh, Stellman and Greene—you’re the Wideband Delphi people? I always use that to estimate!” (Seriously, people actually said that to us!) Or, much worse, they’d tell us, “Your book talks about requirements specifications, and I never use them. But I develop software just fine. You must be wrong!”

Practices are cold. Practices are theoretical. You can sit and talk about the virtues of one or another, and have hypothetical arguments about whether they’ll work in one situation or another. We’ve done our own share of arguing about that ourselves, talking until we’re all blue in the face about when is the correct time to do requirements gathering and how to make a project more or less agile. Those kinds of things are really where people draw battle lines.

But it’s not where the meat of the work happens, honestly. How you make those decisions has an impact on the project, certainly, and sometimes a big one. But it’s not nearly as important as *who* you have on your team: how skilled they are, and how well they work together. That’s when we realized that practices are only one aspect of building better software. And although it’s the aspect we’ve spent the most time studying and dealing with, it took us a long time to realize that it’s usually not the most important aspect.

So, we started looking for ways to get people to open up about their own projects, and to listen to each other about what they’ve learned. We put together a talk about best practices—because that’s what we wrote about and what we knew best—and tried to explain, in as general and non-prescriptive terms as we could think of, how to use them to make your projects run better. We naïvely thought that people would be excited to hear about what we learned, both in writing our book and in running our own projects. We’d talk about the kinds of pitfalls they’d avoid, and we’d give them the tools to avoid them (“Look, here’s one way you can run a productive estimation session!”).

It was a disaster. One talk comes to mind as particularly bad, although not too much worse than many others that we gave. We were invited to give a presentation for a brown bag series at a major Internet company’s New York office. We started in on the talk, going through the kinds of problems that software teams typically face, and outlining the practices that we found to be useful to prevent them. We could see the mounting boredom and even frustration on the faces of the developers. They were mostly young, the majority under 30, and by the time we were halfway through the talk most were checking their PDAs and laptops. Some people even got up and left. And when we got to the Q&A session at the end, we found out where their heads were. One programmer, a large guy with a long gray beard, challenged us when we mentioned agile project planning: “Agile means that you don’t write anything down, you just start coding immediately.” Everyone nodded their heads. We tried to redirect that—“No, that’s not what agile necessarily says!”—but it

didn't matter. Our message wasn't getting through, and we were clearly wasting everyone's time.

We shouldn't have been surprised that we were met with a lukewarm (at best) reception. After all, we'd sat through our own share of networking activities (project management group meetings, architect SIG meetings, software process network meetings...). And for every one that featured a really memorable talk or discussion, there were literally a dozen that boiled down to an advertisement for a consulting company, or a book, or a professional product or service of some kind, and left us completely cold.

So, we reflected on the criticism about agile we'd gotten at that talk. And to our surprise, we realized that he wasn't wrong to criticize us, even if we disagreed with him on the facts. Yes, the criticism stung at first, and we really had to take some time to figure out why the talk ended up the way it did. But in the end, his criticism ultimately showed us what was wrong with our approach. *There is no "best" practice*—at least, none that we know of that will guarantee success every time. Whether a particular tool or technique works depends on the circumstances: the project, the people, and all sorts of mushy, messy stuff that you just can't quantify or prescribe. Now, that's not to say that we don't care about process. We're not distancing ourselves from the practices that we use and that we write about. In fact, just the opposite: we still use them every day, we still write about them, and we still train other people to build better software using them because they really do work—in the right situation, for the right project, and with the right people. But sometimes the way to get something done is to do anything *but* the "right" thing. Sometimes you need to just do whatever works.

Growth for us came from realizing that there really are a lot of ways to run a project. And even though we found some practices that we're very comfortable with, there are times when they're not appropriate. We've had to get a lot more flexible over the years with our own approaches to building software, and how we work with our own teams. The reason we had to do that is because some of our projects failed. Not many, but some. And when they failed, that's when we learned the most.

That's what set us down a new path. We tried to figure out what made some projects work while others crashed and burned. It was hard to figure out one particular cause at first, and we suspected that there was no single answer. And when we started talking to people around us, people we knew, we started to hear the same thing over and over: people talked fondly about one project or another that really stuck with them, and they wanted to share their "war stories" about the terrible manager they had, or the horrible situation they faced, or another problem they overcame. In fact, simply asking people about teams they'd been on often got them to reflect on their own careers, and on their own impact on the people around them. And to our surprise, this seemed to be consistent with everyone involved in every level of software development: from the junior programmer who'd been out of school for only a year or two to the CTO who had worked his way up through the ranks from being a junior programmer years before.

The more we opened up and let other people talk about their own experiences without trying to impose our own practices and views on them, the fewer arguments we got into. And the more we talked this over, among ourselves and with other people, the more we started thinking about four distinct “buckets” (for lack of a better word) that these conversations fell into: *people* (who’s on the team), *goals* (what brings them together), *practices* (how they build the software), and *obstacles* (what gets in their way). This book is organized into parts based on those ideas. We end the book with a section on groups of people that make music, because that helped us see a greater truth about what makes teams work.

Why These Contributors?

We started this project with one core assumption: the more you know about how different people run their projects, the better equipped you are to run your own. We started out by going to people who we knew personally, from our own teams and people we’d met at conferences, and from being around the software development world. We were surprised at the diversity of opinions just among the people who we knew. We were also surprised that we didn’t agree with everything they said—but more importantly, we were surprised to find that we were still able to see the truth in their ideas.

That’s when we came up with a cohesive goal for the book: to draw as many different opinions from as many different areas of software development as we could, regardless of whether they meshed cleanly with our own or even with each other. Yes, this would almost certainly mean that there would be contradictory ideas. But we decided that not only is that OK, but in fact it’s a benefit. (It’s a feature, not a bug.) It showed us that even though ideas are contradictory, they still work in different situations. All projects are unique; all teams are different. It would be very odd if there really were a single answer. There’s room for all of these opinions, and there are situations in which any of them can be the right way to go.

Now, that’s not to say that there are no wrong answers. In fact, just the opposite is true, which is why we included stories in this book about the teams that are *not* successful. Just as the right decision at the right time can make all the difference in a successful project, sometimes people can be truly misguided when they build software. They choose the wrong practice or path: for example, good documentation can turn bad when it’s onerous and not used. But projects can fall off the other side of the spectrum, too, where people on the team don’t plan at all, and everyone goes in a different direction. And that’s what the cautionary stories in this book demonstrate: teams that started out with the wrong goal, included the wrong people, applied the wrong practices, or simply hit an insurmountable obstacle.

We could never have written a book that does that by ourselves. We’re very limited in what we know. Everybody is. And that’s why we cast as wide a net as possible, talking to people whose opinions we respect. The many people who contributed to this book represent the real spectrum of projects that people across the whole industry take on.

And while many of the opinions and ideas that you'll read about differ from our own—in some cases, they're completely opposite—we learned a lot from each and every one of these contributors. We think that you will, too.

This book includes stories and interviews with veteran team leaders from all around the software industry. We recruited contributors from as many different industries and areas of interest as possible: from defense to social organizing, from academic research to video game development, from aerospace and defense to search engines, and from project managers to “boots-on-the-ground” programmers and system admins. There are people who we met over the course of our educations and work lives. There are contributors from a wide range of companies, including people who worked (and, in some cases, still work) at NASA, Google, IBM, and Microsoft. We were especially surprised and pleased when we got contributions from people like Grady Booch, Barry Boehm, Steve McConnell, and Karl Wieggers, whose writing and teaching were central to our own understanding of how software is built. We felt honored to work with them and other people in this book who you may not know as well, but who are also doing amazing and innovative things for software development. We're especially grateful for contributions from Tim O'Reilly (whose publishing company printed this and our other books), Scott Berkun (who we've known for years and who not only contributed a great essay, but also interviewed Steve McConnell), and Tony Visconti (a legendary music producer who took the time to talk to us about his own process).

Frankly, we're still amazed that we were able to get such a wide-ranging, knowledgeable, distinguished, and talented group of contributors. But even more surprisingly, not a single contributor asked to be paid. Instead of dividing the royalties from this book among the contributors, we have the privilege of donating them to PlayPumps International, an innovative charity that digs wells to deliver clean drinking water to schools and villages in rural sub-Saharan Africa. PlayPumps is more than a charity; they've had to do their own share of engineering and innovation. You'll learn more about them and their mission (and something about teams, as well!) in our interview with Trevor Field, the founder of PlayPumps.

Every one of these contributors has something interesting, important, and, most significantly, *useful* to say about teams: how they work, how to build them, and how they break down. Each of them is a veteran team leader in his or her own right, with successes and failures under his or her belt. In some cases, we were surprised and even shocked by the stories they had to tell. And every single one is entertaining. The sum of all of these parts amounts to an experience that no single person could cobble together.

Before we delve into the four areas we chose to examine—people, goals, practices, and obstacles—we'd like to start with our interview with Tim O'Reilly because he touches on themes that you'll see over and over again throughout the book. As you're reading it, keep an eye out for each of those four things. When we talked to Tim, he was able to make a lot of these concepts concrete for us, putting words to a lot of the ideas that had been swirling around in our heads after editing the stories and interviews in the book.

We weren't really sure what to expect when we interviewed Tim O'Reilly. Between the two of us, Jenny had met him only once, at the Amazon Developer Days conference (where she'd been invited to speak), but we'd read interviews with him, and his writing on O'Reilly Radar. We were particularly interested in what he had to say about leadership and how to direct groups of people without micromanaging them.

So, if you picked up this book hoping to find the *One Correct Way*™ to run a beautiful team, we're really sorry, because that's not what this book is about. But if you're looking to gain some insight into what makes a good team tick, and what you can do to take a mediocre team and make it better—or take a great team and make it crash and burn—you're going to get a lot out of this book. We think that it's a good read, and at times a gripping one. It represents a wealth of experience, from a very wide range of people all around the industry (and a few who aren't in our industry at all). It attempts to find the general things that hang teams together, without giving you prescription, dogma, theory, or overt advice. We hope you enjoy reading *Beautiful Teams* as much as we've enjoyed bringing it together.

—Andrew Stellman and Jennifer Greene
March 2009

Preface

BEAUTIFUL TEAMS WAS CONCEIVED IN LATE 2007 DURING A CHANCE MEETING IN O'REILLY EDITOR

Andy Oram's office in Cambridge, Massachusetts. We'd been invited to give a talk for the local PMI chapter, and we decided to take the opportunity to drop by the O'Reilly office to say "Hi" and finally put faces to some very familiar voices we'd worked with over the years. *Beautiful Code* had spent a few months at the top of the O'Reilly bestseller list, and the company was looking to follow up with another anthology. Since we've spent so much of our careers talking and writing about how projects work and how teams build software, the idea for *Beautiful Teams* basically fell out of thin air.

The original idea was just to follow up on *Beautiful Code* with a straightforward anthology about project management. Like all great projects, *Beautiful Teams* took on a life of its own. It attracted contributors of an incredibly high caliber. It became a journey for us, allowing us the opportunity to learn from some of the brightest minds in software development today. These are personal stories and experiences. Each person who contributed to this book is talking about his or her own past work life, which very few of us ever get a chance to examine. And every single contributor was happy to donate his or her time and effort without any payment whatsoever; proceeds from this book are instead being donated to PlayPumps International.

How This Book Is Organized

Here is a short summary of the chapters in this book and what you'll find inside:

Chapter 1, *Leadership, an Interview with Tim O'Reilly*

Tim O'Reilly gives us his thoughts on leading teams and companies, and moving the world of software forward.

Part I, *People*

Chapter 2, *Why Ugly Teams Win, by Scott Berkun*

Many so-called beautiful teams were never described in those words by the people on them. Scott relates his experience at Microsoft, and explains the *wabi-sabi* of ugly teams.

Chapter 3, *Building Video Games, an Interview with Mark Healey*

The cofounder of Media Molecule talks about what he learned building the hit video game LittleBigPlanet.

Chapter 4, *Building the Perfect Team, by Bill DiPierre*

Bill tells the story of how a good manager can take a disparate group of people and turn them into a great team.

Chapter 5, *What Makes Developers Tick, an Interview with Andy Lester*

Programming manager and Perl contributor Andy explains what motivates developers and how they can improve their relations with their teams.

Chapter 6, *Inspiring People, an Interview with Keoki Andrus*

Keoki tells us about how he has improved teams in companies such as Intuit, Microsoft, and Novell by understanding, inspiring, and guiding the people on them.

Chapter 7, *Bringing the Music Industry into the 21st Century, by Tom Tarka*

This is the story of the rise and fall of MP3.com, an icon of the dot-com boom and bust, and the people who lived through it.

Chapter 8, *Inner Source, an Interview with Auke Jilderda*

The inner source initiative brings open source practices and ideas to corporate teams, and Auke tells us how he implemented it, and how it affected the people on those teams.

Part II, *Goals*

Chapter 9, *Creating Team Cultures, an Interview with Grady Booch*

It takes work to get a team to gel, especially a distributed team. Grady talks about the challenges of getting teams moving in the right direction.

Chapter 10, *Putting the "I" in Failure, by Jennifer Greene*

Jennifer tells the story of her experience working on a great team with conflicting goals.

Chapter 11, *Planning, an Interview with Mike Cohn*

Through stories from his own career, Mike tells us about how understanding the context around a project means the difference between succeeding and failing.

Chapter 12, *The Copyfighters Take Mordor, by Cory Doctorow*

This is the story of how a great team that's motivated by social responsibility can succeed against a daunting foe.

Chapter 13, *Defending the Free World, an Interview with Neil Siegel*

The CTO of a major defense and aerospace company tells us about how he motivates his software developers.

Chapter 14, *Saving Lives, an Interview with Trevor Field*

The founder of PlayPumps International talks about what motivated him to leave the cushy world of advertising and dedicate his life to delivering clean drinking water to rural schools and villages in sub-Saharan Africa.

Part III, *Practices*

Chapter 15, *Building a Team with Collaboration and Learning, by James Grenning*

An early agile trailblazer and signer of the Agile Manifesto, James talks about his first experience with agile methods.

Chapter 16, *Better Practices, an Interview with Steve McConnell*

Scott Berkun and Steve McConnell discuss how better development practices can lead to high-performance teams.

Chapter 17, *Memories of TRW's Software Productivity Project, by Barry Boehm and Maria Penedo*

This is the story of one of the first successful process improvements ever done, told by a pioneer of the industry.

Chapter 18, *Building Spaceships, an Interview with Peter Glück*

Peter talks about the challenges of building software that will be shot into space at NASA's Jet Propulsion Laboratory.

Chapter 19, *Succeeding with Requirements, by Karl E. Wiegers*

Software requirements can make or break a project, and Karl uses them to ensure success.

Chapter 20, *Development at Google, an Interview with Alex Martelli*

Alex explains how better planning and agile practices improve life at a cutting-edge company.

Chapter 21, *Teams and Tools, by Karl Fogel*

Karl shows how a software tool can have an enormous impact on the way a team works.

Chapter 22, *Research Teams, an Interview with Michael Collins*

Michael tells us about his work on a security research project.

Chapter 23, *The HADS Team*, by Karl Rehmer

Building flight software for the Boeing 777 required a whole new set of tools to be written, which brought its own set of challenges.

Part IV, *Obstacles*

Chapter 24, *Bad Boss*, by Andrew Stellman

One bad manager can destroy a team.

Chapter 25, *Welcome to the Process*, by Ned Robinson

A good team can overcome even the most incredible and unforeseen challenges.

Chapter 26, *Getting Past Obstacles*, an Interview with Scott Ambler

Lots of different problems can trip up a team. Scott tells us how to get past some of the biggest ones.

Chapter 27, *Speed Versus Quality*, by Johanna Rothman

A new project manager faces stiff challenges when she joins her team.

Chapter 28, *Tight, Isn't It?*, by Mark Denovich and Eric Renkey

An improbably great team faces obstacle after obstacle.

Chapter 29, *Inside and Outside the Box*, by Patricia Ensworth

A team that's faced with poor management, terrible facilities, and interpersonal problems manages to stay together despite it all.

Chapter 30, *Compiling the Voice of a Team*, by Andy Oram

One developer can take on management when the facts are in his favor.

Part V, *Music*

Chapter 31, *Producing Music*, an Interview with Tony Visconti

Legendary record producer Tony Visconti shows us that producing records and building software have a lot in common.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/9780596518028>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://www.oreilly.com>

Safari® Books Online



When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at <http://my.safaribooksonline.com>.

Acknowledgments

We'd like to thank all of the contributors who generously donated their time and effort: Tim O'Reilly, Scott Berkun, Mark Healey (and the rest of the folks at Media Molecule!), Bill DiPierre, Andy Lester, Keoki Andrus, Tommy Tarka, Auke Jilderda, Grady Booch, Mike Cohn, Cory Doctorow, Neil Siegel, James Grenning, Steve McConnell, Barry Boehm, Maria Penedo, Peter Glück, Karl Wieggers, Alex Martelli, Karl Fogel, Mike Collins, Karl Rehmer, Ned Robinson, Scott Ambler, Johanna Rothman, Mark Denovich, Eric Renkey, Stan Granite, Patricia Ensworth, and Andy Oram. And we are especially grateful to Tony Visconti for taking the time to talk to us.

We want to thank Trevor Field and the rest of the great people at PlayPumps International (<http://www.playpumps.org>) for the wonderful work that they do, as well as for Trevor's contribution to this book. Please visit the organization's website to find out why we're donating royalties from this book to them.

Finally, once again, special thanks to Nisha Sondhe for her extraordinary photography.

About the Editors



Andrew Stellman started work as a programmer at a national record label after graduating from the School of Computer Science at Carnegie Mellon University. He spent the first few years focusing on technical design and architecture, but found himself increasingly responsible for leading and then managing the teams he was on. Over the last decade, Andrew has led teams of programmers and managed numerous projects. He now spends his time building his own projects and helping the people around him improve the way they manage theirs.



Jennifer Greene began her software engineering career in the early 1990s at a well-known online service. She has since built and led software engineering teams in organizations spanning finance, non-profit, natural language processing and media. She has built and managed distributed software teams and worked with developers around the world to improve their practices and build better software.

Jennifer and **Andrew** have been working together since 1998 building software and training software teams. They initially defined the software process at the company where they were working, and have since run many successful projects together. Their projects have been focused on science and public health, and they have built software for the Mailman School of Public Health at Columbia University, the Business School (also at Columbia University), and the National Academy of Sciences. They have since branched out into commercial project management and software engineering consulting, providing training, project management, outsourced project management, and professional services to both the business and academic communities.

Their first book, *Applied Software Project Management*, was published by O'Reilly in 2005, and has since been adopted as a software engineering textbook at many universities worldwide. They went on to write two books in the acclaimed Head First series. *Head First PMP* has been recognized as one of the best PMP preparation guides available, and has been used by tens of thousands of project managers to prepare for their project management certification. Their third book, *Head First C#*, spent more than a year as one of O'Reilly's bestsellers and is currently one of the top-selling programming language books on the market. In addition to writing, Andrew and Jennifer speak regularly at software development, project management, and process improvement conferences around the world. *Beautiful Teams* is their fourth book.

CONTENTS

| | | |
|-----------------|---|------------|
| | WHY BEAUTIFUL TEAMS? | xiii |
| | PREFACE | xix |
| 1 | LEADERSHIP <i>an Interview with Tim O'Reilly</i> | 1 |
| Part One | PEOPLE | 11 |
| 2 | WHY UGLY TEAMS WIN <i>by Scott Berkun</i> | 15 |
| 3 | BUILDING VIDEO GAMES <i>an Interview with Mark Healey</i> | 21 |
| 4 | BUILDING THE PERFECT TEAM <i>by Bill DiPierre</i> | 33 |
| 5 | WHAT MAKES DEVELOPERS TICK <i>an Interview with Andy Lester</i> | 47 |
| 6 | INSPIRING PEOPLE <i>an Interview with Keoki Andrus</i> | 59 |
| 7 | BRINGING THE MUSIC INDUSTRY INTO THE 21ST CENTURY <i>by Tom Tarka</i> | 75 |
| 8 | INNER SOURCE <i>an Interview with Auke Jilderda</i> | 103 |
| Part Two | GOALS | 113 |
| 9 | CREATING TEAM CULTURES <i>an Interview with Grady Booch</i> | 117 |
| 10 | PUTTING THE "I" IN FAILURE <i>by Jennifer Greene</i> | 127 |
| 11 | PLANNING <i>an Interview with Mike Cohn</i> | 135 |
| 12 | THE COPYFIGHTERS TAKE MORDOR <i>by Cory Doctorow</i> | 153 |