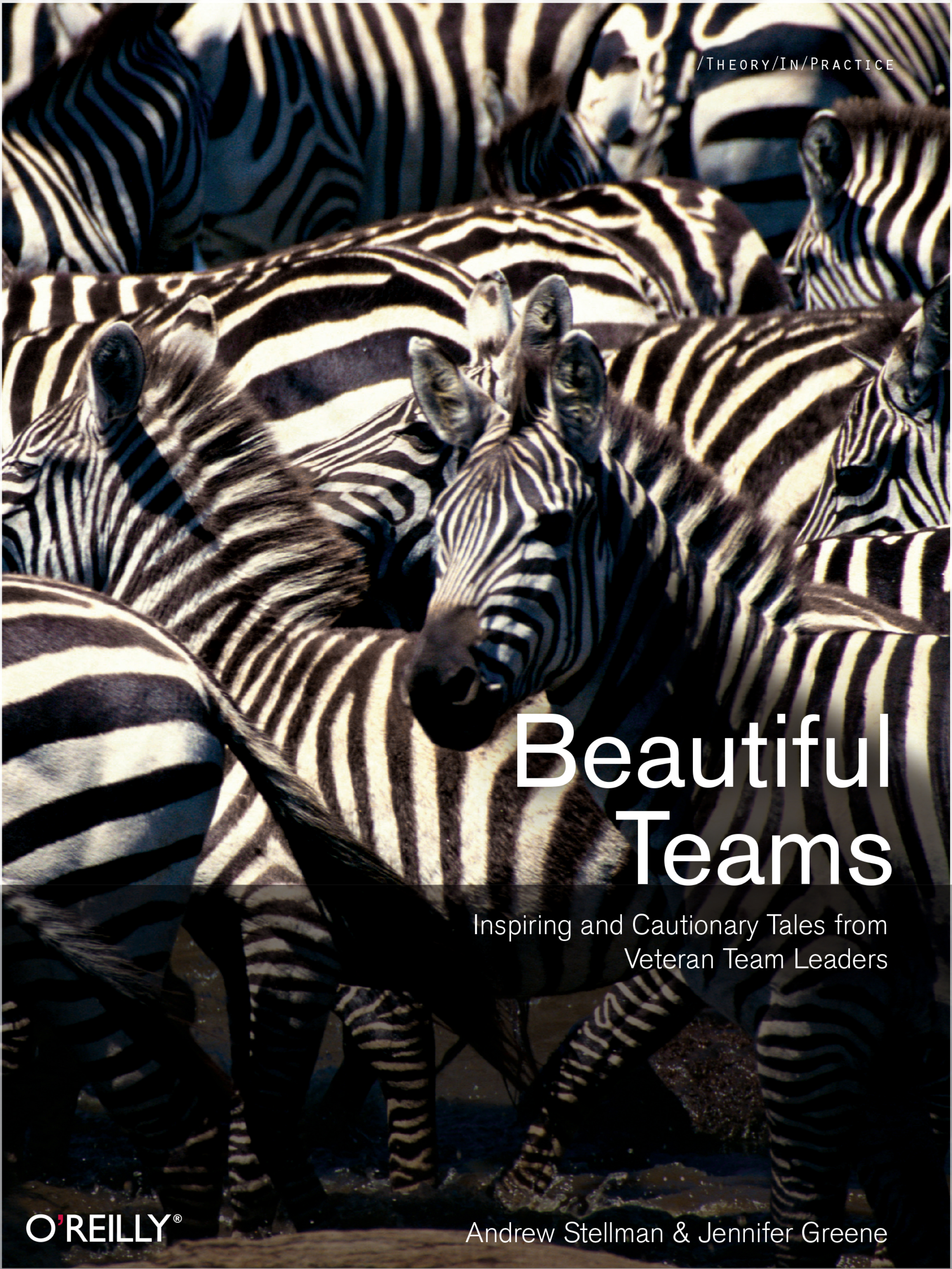


/THEORY/IN/PRACTICE

Beautiful Teams

Inspiring and Cautionary Tales from Veteran Team Leaders

Andrew Stellman & Jennifer Greene

O'REILLY®

Praise for Beautiful Teams

"Stop complaining about your coworkers. Instead, get your team and your boss to read *Beautiful Teams*. It proves the the amazing potential of teamwork and how it can happen in your company."

—SCOTT BERKUN, BESTSELLING AUTHOR OF *THE MYTHS OF INNOVATION* AND *MAKING THINGS HAPPEN*

"The striking diversity of these histories, experiences, and reflections tells a broader, deeper story than any single author could manage. Somewhere in this book you'll certainly find key tips for dealing with the very team-building and team-running issues you're having today, as well as outlook-broadening views of issues that are quite outside your experience so far."

—ALEX MARTELLI, GOOGLE

"In *Beautiful Teams*, the many contributors step back from advancing their usual prescriptions to celebrate their own successes (and yes, challenges) within teams. In this celebration, they provide some of the best insights that we can carry forward into our own careers. Jennifer Greene draws wondrous team memories from the ashes of a dot-com failure, Keoki Andrus shares a healthy respect for innovation and creative play to inspire a team, and there are engaging stories by Karl Wiegers and many others that capture great team experiences. *Beautiful Teams* will keep you rapt like few technical books ever have."

—JIM BROSSEAU, AUTHOR OF *SOFTWARE TEAMWORK: TAKING OWNERSHIP FOR SUCCESS*

"Teams build software, and if your team isn't working together effectively, you have very little hope of success at the software development game. The experiences and insights shared in this book will stick with you for years to come."

—SCOTT W. AMBLER, WORLDWIDE PRACTICE LEADER AGILE DEVELOPMENT, IBM SOFTWARE GROUP, AND AUTHOR OF *AGILE MODELING*

"The good, the bad, the ugly, and of course, the beautiful. You'll read about teams who persevered no matter what, and how some famous organizations built teams. If you want to know how your team stacks up, or how to make it beautiful, read this book. The stories are captivating and all too real."

—JOHANNA ROTHMAN, CONSULTANT AND AUTHOR

Beautiful Teams

Edited by Andrew Stellman and Jennifer Greene

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

PART FIVE

Music

We started this book by interviewing Tim O'Reilly because we felt that he covered a wide range of topics that we'd delve into later on. One of the things that Tim did was draw a parallel between teams of developers and how groups of musicians work together. That piqued our interest, and we wanted to dig a little deeper.

Tim: I have to say that my greatest weakness as a team leader, to be quite honest, has always been that I've always taken what's available and tried to make something of it, rather than an engineering approach—we need one of these, and one of these, and one of these—and building something out of nothing.

Jenny: That's actually my question. As you were talking, one of the things you said really resonated with me. The traditional way to handle deciding what you're going to build is getting everybody to agree on that up front. It sounds like you're more interested in the discovery process, allowing people to come to whatever truth comes from the experience. How do you reconcile that?

Tim: Well, yes and no. It's a good question. It's so paradoxical, because it sounds like when you say whatever truth comes from the experience—that sounds very California and New Age. When I say, "uncover the statue in the stone," it's really there. There's not more than one. Well, maybe there's a couple, but it's not like any old thing works. It's not like you can just go with the flow and do what you feel. There's an aesthetic vision that drives the process.

Jenny: So, the leader has to carry that vision?

Tim: Yes, absolutely. You have to know what you're going for. And what you're doing is going back to the idea of humming a tune and getting other people to follow it. Well, there is an idea there of harmony. There is an idea there because there is a tune, and somebody can be off-key. It's not like everyone can go hum their own tune. This is the tune. Find it, converge. And that's the skill in bringing a creative team together.

Going back to the analogy of humming the tune, and having people learn to play it on their own instruments: if people are accomplished musicians, they can do that. Then they can actually start to elaborate on the tune, they can build on what you've done. But it starts what in open source what Eric Raymond calls a "plausible promise." There is an aesthetic vision there where people say, "Yeah, I get it, I want to be part of that." Then they can express their own creativity. And if you've done it right, you haven't overspecified, you haven't told people what to do or how to do it. You've just given them a vision of where you're going, and they find their own way there.

Andrew: OK, I've got a question. There are definitely some really skilled, really amazing band leaders. Like, say, Count Basie, who's famous for not using as much sheet music and charts, and really having musicians who work exactly like that. But I'm sure history is littered with the names of bands we don't know with people who tried to work like that

and weren't able to. And that's one thing that we've seen a lot in software projects specifically. You'll have really talented people, and a leader who's really smart and talented, yet the team has a lot of trouble getting software out the door.

Tim: And a lot of what I've done in my career has been to tell stories that help people to organize their own activity. So, I'm a very non-directive leader or manager. In fact, I often joke that I'm kind of like the title character in the '50s musical *The Music Man*. You know the story, the guy was a con man who sold instruments, and said that he'd teach the kids to play music. But of course, he didn't know anything about music. He said he had this new system called the "Think System," where he'd hum the tune and the kids would figure out how to play it on their own instruments. And of course, it works out in the story; the kids do kind of figure it out, and it works out just in time to save him because he's fallen in love with the town librarian.

Andrew: Could you run a real band like that?

Tim: Well, you could, of course, if you had very skilled players. In fact, that's what jazz is all about.

Andrew: I guess you're right—that's pretty much how Count Basie ran his band.

Tim: A jazz band is the Think System applied. Somebody puts out an idea, and someone else picks up on it. I still remember, actually, when Jeff Bezos and I talked to various congressional people about the patent system. One of the congressmen, well known; we were trying to make the analogy of how invention works, and that it's a little bit like jazz. And he totally got it, because he was a big jazz fan. "Oh my God! What if all of those jazz riffs were patented, and you couldn't take them and run with them?"

So, I think in a lot of ways that gets to the heart of something. If you have good enough people who are motivated and excited and skilled, you can in fact hum a tune and have them pick up and improvise on it. And that's what I've done through a whole lot of my career. That's not to say there isn't a lot of hands-on. But even that is in the form of storytelling. The team of editor and writer, for example, introducing a book: the way that I would often do it is that I would rewrite something and show them, and then they'd say, "Oh yeah, I get that now." As an editor, that's also how I'd work with assistant editors—I'd kind of demonstrate stuff, and they'd demonstrate back what they took away from me. And then, when I said, "Yeah, you've got it," I'd have less and less to do.

I know I've said two very contradictory things here, because I said, "finding the statue that's in the stone," but I also said, "articulating a vision that other people can believe in." One of those things sounds like there is the ideal form sitting there in the form, where there's only one. In the other, you can make anything out of anything. And the truth is somewhere in between. What it feels like from the inside, when you get it right, is like doing harmony in music. There are a lot of different harmonies possible, but you do have to converge on something.

So in some sense, I think that analogy is probably a pretty good one. There are some underlying things that make a situation work, that make it come together. That's something else that's a big part of my thinking—Alfred Korzybski, with general semantics: "The map is not the territory." The map matches the territory. And when you create an aesthetic vision that somehow matches a territory, that somehow makes the thing more what it is, then you get people to follow you. And when you seed that moment, when you're able to articulate a vision, you get people to sign up for it. And together you build something that is true, and feels true.

As we were editing our interview with Tim, we felt that this one section really stood on its own, because it said something to us. We felt that it made a poignant statement about how teams can revolve around a single visionary leader, and that one leader can bring everyone together under an overarching vision. But he said something else that struck us: in a way, he erased the line between a vision for a project and an artistic vision.

We felt that these ideas needed more exploration. So, we sought out an expert in how musicians work together, and we were thrilled that Tony Visconti could take the time to talk to us. Both of us have been fans of a lot of his work over the years, even when we didn't necessarily know that he was behind the music. Tony has spent his entire career figuring out how to get groups—teams, if you will—of musicians to work together.

We weren't sure exactly how this would turn out. But what really took us by surprise is just how much of what Tony said echoed many of the things that our other contributors said throughout the book. While you're reading this interview, keep your eyes open for them: see if you can spot ideas, sentiments, and opinions that reflect other things you've read over the past few hundred pages.

CHAPTER THIRTY-ONE

Producing Music

an Interview with Tony Visconti

If you don't know Tony Visconti's name already, just go to his website (*http://www.tonyvisconti.com*) and click on the Discography link. If you've been alive in the past four decades, you almost certainly own at least a few albums that he's produced (including some of our own favorites). He's worked with some of the greatest stars—and biggest egos—in the music business, and he's created lasting albums that were both artistic and commercial successes. We wanted to know how he did it…and to our amazement, he sat down with Andrew and told us! And it turns out that what he has to say is surprisingly relevant to the way we build software.

Andrew: Thanks so much for taking the time to do this. Since you're a producer, I want to start out by talking about just what it is that you do. Because I think a lot of people don't really have any idea what a producer does. There's an old Rocky and Bullwinkle episode where Boris Badenov hires Bullwinkle as a movie producer. Bullwinkle asks Boris what a producer produces, and Boris says, "Money." I've definitely noticed that people who have job titles like "producer," "art director," "editor," and "project manager" seem to have a lot in common. So, how would you describe what you do?

Tony: Well, those job definitions are very vague at best, because what someone brings to a job needs to be defined. My style of production is different from, say, a Hugh Padgham style of production, or even George Martin's style of production. A producer of a record is more like—well, one of the jobs is money manager. We don't raise the money, but in the old days, producers used to raise the money. Nowadays, the record company is the actual producer of an album or a project.

They'll say, "We have so much money and we'd like you to make us this kind of an album." So, they define the nature of the job in the first place. I, as the producer of the record, manage the entire production of the record from the moment it's handed over to me to the moment it's delivered. The responsibility is mine.

Andrew: That sounds a lot like what a project manager does. Or what a team leader does, in a lot of ways—but when a lead developer is doing that, he's basically wearing a project manager hat.

Tony: Yes, I guess that's a project manager—it's probably the similar thing in your world.

Andrew: And what you said about vague job definitions definitely makes sense in this case. Ask any programmer on a team, "What does a project manager do?" If they can come up with an answer at all, what you said is pretty close to what they would say. And I suspect that if you ask a musician in a band working on an album, they would have a lot of trouble saying exactly what it is a producer does, too, except that he sometimes tells you what to do and disagrees with you a lot.

What do you think qualifies someone to be a good producer, or an effective project manager?

Tony: I believe that if someone is going to be a project manager, they should have very intimate knowledge of all the jobs that they are going to manage. All the different stages, all the levels of work that needs to be done, like differentiate between the creative team and the logistical team. I mean, there are members in the team who are responsible for creativity, and members who are responsible for logistics, how to proceed, that things have to be done in an orderly way. So, a producer should know how to do these things.

I think a producer—or a project manager—should come up from the ranks, and should have had practical day-to-day work on all the jobs they are about to manage. Otherwise, they really won't understand the situation, and they'll be at the mercy of a person who's trying to get away with murder, someone who's trying to shirk their responsibilities. I've done every single job in the music business. I've been signed as a recording artist.

I've played bass. I've played guitar. I've played piano. And at one point I learned how to become an engineer. I learned how to orchestrate and write for orchestras—actually write real music on real paper and have musicians play off the paper.

Nowadays, I don't actually do all of those jobs at the level I used to do, because they were solitary jobs. I would be a bass player in one situation or I'd be an orchestrator in another. I can manage a team full of people in a recording situation because I know exactly what it takes to do those jobs.

Andrew: But you wouldn't necessarily be the best person for any of those jobs, better than a better musician or a better orchestrator or a better engineer than the people you're working with. You don't need to be the best, just have a good working knowledge of each job.

Tony: As years go on, I find that I don't want to do those jobs because it's just too much to do all those things at the same time. But when you have a lower budget, people will have to double up on jobs. For instance, if the budget is so small that I can afford only one assistant and we have to record a band, then I will take on a couple of jobs involved there. I'll be managing myself, in a way. But ideally, you want to have a team of experts. I know I'm a good bass player, but I want someone who's actually a great bass player.

I know what I can expect of a person on my team because of my experience. I know what's possible, and often my requests verge on the impossible. I can get performances out of musicians only because I know, I just know what can be done. I don't necessarily have to be better than them, but I think I'm qualified to direct them and to coach them because I've done that job.

Andrew: So, it's almost like a respect thing. Someone's really only going to take your direction or take your advice if they respect your ability, at least respect that you know enough to give them that kind of direction.

Tony: Yes. It's akin to martial arts. You won't respect somebody who's been doing martial arts for six months to teach you. Even as a novice, you can see that their body is uncoordinated and they can't really do what they're talking about. But when you study with a great martial arts teacher who's practiced for 35 years, then there's respect; you will do anything that person says. And if that guy's really experienced in everything he talks about, you have to respect that person. And he will respect you, because you come to him to learn or to achieve something.

So, that's the way I see it. I don't know if it sounds arrogant, but this happened to me quite early on because I studied music at a very early age. I would say I was qualified to be a record producer by the time I was 25 years old. The thing I did not have was people skills. For me, that came out of the school of hard knocks. Learning how to speak to people is actually a very important lesson to learn. How to actually communicate to people without anger, without sounding dishonest, disingenuous.

Andrew: What happens if you don't communicate well with people, with respect or honesty? What if you're transparent or disingenuous? What goes wrong on your team, with your project?

Tony: Well, if you don't correct problems directly and efficiently, then you'll lose their interest. They'll lose respect for you, and they'll lose interest in what you say. They'll think you don't know what you're talking about. That's the worst thing in the world to happen to a producer or project manager—where you're perceived as some kind of a dork who hasn't a clue. You don't ever want that to happen. You've lost the team once that happens.

And I'll tell you, if you do happen to put your foot in your mouth, a way to regain the respect of the team is to simply say, "I've made a mistake, could we start again on this? I've given it a lot of thought and I made a judgment error." That's how you gain the respect of a team again after you've lost it: you have to admit that you've made a mistake. Your honesty and courage to admit mistakes will maintain your team's respect. If you try to cover up your ignorance, or if you try to look cool all the time, you just get deeper and deeper into phoniness and your team will drift away from you.

Andrew: I definitely see really similar things on programming teams. It's not enough for a team leader to be respectable. You have to have—I don't want to call humility, but at least a willingness to see that somebody else is right.

Tony: Yes.

Andrew: And I know a lot of programmers have trouble with that. I'm assuming that a lot of rock stars do, too.

Tony: There're always going to be differences of opinion, but the hierarchy is always the same. The person who is managing the project is in charge. That always has to be clear and understood. But that does not give you a license to be a tyrant. One of the most positive goals of a good project manager is to bring out the best in the people in the project, not to stifle them, not to hold them back, not to squeeze them between two walls. It's very dangerous to do things like that. What you want to do is to create heroes on your team, to make people on your team—well, all of them if possible—make them shine, to really make your team look like the A team.

And you're not going to get that if you don't listen to them. In order for them to be members of the A team, you have to let their opinions be heard; you have to, you must. This also means that you have to be a flexible thinker and able to modify the goal of the project. If some maverick kind of idea comes up that can be incorporated—some stroke of genius comes from a team member—you have to be open to that. Even if it seems like a ridiculous idea at first, don't make snap decisions. Don't say, "Oh no, this isn't in the plan, it's not going to work." Take time to consider all ideas.

If a maverick idea turns out to be a bad idea, don't humiliate the proposer. You have to get across to your team that an idea is only an idea and they arise spontaneously. You don't want to make it personal when you're rejecting someone's idea. You can sincerely say it was a good idea, but not really germane to this project. There are ways of communicating, where you bring out the best in someone and you don't stifle them when you really feel that they're going down the wrong path.

Andrew: What happens if you feel the whole project is going in the wrong direction? You know you need to get it on the right track, but you just have a feeling that a lot of people on the team are going to be hard to convince. Even if they'll be happy with the final product after you change direction, the idea of making the change just isn't going to sit well with them. How do you handle it if there's just something that they don't want to do, some change that you know is going to meet with resistance? You have to be the bearer of bad news. How do you handle a situation like that?

Tony: Well, the nature of my business is it's very visceral. Music is an emotional product. People laugh and cry and make babies and dance to music. Music serves a lot of purposes in life, and it's very, very emotional. And I think everyone has a common feeling when they're making music. You kind of feel it in your gut that things are going well. I mean, it's very, very clear after a playback, when somebody's played really well—they come in and usually you see smiling faces. If you don't see smiling faces, and you yourself are not smiling, then something is going wrong. That's all I think a project manager has to say: "We're doing it wrong. We have to go back to the drawing board."

Don't moan about lost time, because the important thing is that the team has derailed and has to get back on the right track. I feel that way; I mean, I've stayed up all night recording a track, a vocal, or something. We'd come back the next day and we think it should be good because we stayed up all night. But you have to tell the truth—is it really good? I find in moments like these there is a common experience in the team. These harsh moments are easier to take after the team has had a good night's sleep.

It's simple: it's either good or it's bad, and somebody has to say that. I'm a firm believer in honesty when that happens. When that takes place, then I think people do fall into line.

Andrew: Have you ever been in a situation where you feel the team has that unity, but the final product just isn't any good? Everybody is happy with how it came out. But somebody's paying for this—a record company—and when you bring it to them you proudly play the album for whoever it is you're dealing with there, you can see on their faces that this is exactly what they were not looking for, and clearly they don't think they can sell this.

Tony: Yes.

Andrew: How do you handle a situation like that? Say they tell you you're going to have to take this back to the drawing board, and you've got another couple of months and enough budget to do it right. But you know that the team is just not going to want to hear that. Nobody wants to be told to go back to the drawing board after they think they just nailed it. So, does that happen? How do you deal with it?

Tony: It often happens if you've waited too long to present it to the person who commissioned this work. If you leave that right to the end, then you are going to risk that. Your employer doesn't want to know how late you stayed up. They don't want to know about your little internal squabbles. That's nothing. If your results are not what they expected, they'll say to you, "Is this what you brought me after all this time? Is this what we paid for?" Now, this is a very tough situation. In some way, you have to convince your employer that the results are "just different," and you have to tell your team that some modifications are necessary.

I have turned in some albums that were all but rejected and reluctantly released to the public. Sometimes, you're just ahead of your time. An artist person I produced was about three years ahead of his time when he was younger. He wasn't in sync with what the labels or the public wanted. You might hand in what you consider to be your greatest work ever, but by the time it's released it is met with blank stares. The trouble was my artist wasn't like anyone else. He was too iconoclastic, not trendy at first.

I've learned two ways to avoid that great misunderstanding at the end happening. Halfway or maybe three-quarters of the way through the project, I bring what I have to the record company and ask them, "What do you think? Are we going in the right direction?" Some artists are loathe to do that, but it's really important before you go too far. You have to get it out of your head that the people paying for the whole shebang are the enemy.

Psychologically, it's going to go a long way, because if the people who commissioned it feel included at an early stage, they will be more receptive at the end if they feel involved in the middle stages. I didn't like to do that in the old days. I liked to wait until the finished product, because I was quite young and arrogant. But really, you need feedback, and that feedback goes both ways—because what you're telling the client is, "I respect your opinion, and I'd like you to tell me what you think of the project so far." So, that's one way I avoid having that ego clash at the end.

Andrew: I like that idea a lot. The idea of an iterative process where you do part of the work, until you have something that sounds sort of like a finished partial product. Once you've gotten to a milestone—something finished enough that you can bring it to the client—you say, "This is what we've got so far. Here, are we going in the right direction? We're at a point where we can change direction if we need to." Then they can get their fingerprints on it, feel some sense of ownership of the final product. If you do that several times, iterate through the project that way, you'll zero in on something that everybody can agree on.

Tony: I wonder if your programmers have a similar attitude to some rock stars I've worked with, who feel that the record companies are corrupt and stupid and they wouldn't know what a hit record sounds like if it ran them down. But the record producer (or project manager) has to be the person who floats between those two worlds. The record company, they're all about the bottom line, the cash, whereas the programmers or the artists are all about culture and innovation and art and creativity. The project manager has to be the filter between the two worlds, because if the programmers communicated directly with the record company, they'd be thrown off the balcony. They'd be fired for insolence.

And if the record company were allowed to talk to the artists (the programmers), then the artists would throw them off the balcony. You have to understand the way both people's minds work. It's definitely a left-side/right-side brain situation, and that's where a project manager could really shine with their people skills. As I've said, you have to have technical skills. You have to understand the programmer's problems. But on the other hand, you have to understand the company's bottom-line problems, and what they want to sell to the public. So, there you go.

The second thing I do is I always budget in more than I need. It's a little trick of mine. And I book in maybe three extra weeks that I probably won't need, and then what I end up doing is I deliver the project under budget and early.

Because I really know deep down inside how long it's going to take—and I've only learnt this from experience, of course. This way, if you need those three extra weeks, it's available, and it kind of makes everyone breathe a little easier. If you budget just how long it's going to take and you have no contingency, then there's no Plan B. You're setting yourself up to fail. If you come to that crunch, where the client does not like your product and there's no budget left, you have failed, and everyone's going to look bad.

Andrew: So, what happens when your team members just don't get along? What happens when you've got conflicts? One person wants to do it one way, another person wants to do it another way. You've got irreconcilable differences, and the clock is ticking and the budget is burning.

Tony: Well, if they don't get along, that's something that should have been observed in the first place. I pick my team members very carefully. I have a lot of people in my phone book that I use. I know certain drummers will not get along with certain singers, I just know that. For instance, for this new project I'm working on, this drummer was recommended, but he was kind of, well, I won't say any names, but he was kind of a beer-drinking, happy-go-lucky guy who's always telling fart jokes. He plays great drums, and all that. But the singer was the complete opposite—a very sensitive person, a person who doesn't like gross people. It would have been insane to put those two people on the same job. And this drama would be taking the wind out of this performer's sails all the time. So even though I like this guy's drumming, he was the wrong drummer for this artist. The sessions would have collapsed into a kind of men's club, and the artist I'm working with is a very sensitive female.

I try to pick the team members for personality. Personalities are very, very important. So, the personalities have to be right, and of course, the style has to be right—not that everyone has to be in the same style, but the styles have to be complementary. It could be a yin-yang thing. Maybe that's the right thing to do. Maybe that's the wrong thing to do, putting such extreme types, personality types, and creativity types together.

But that has to be worked out, and that's your responsibility as a project manager. Learn as much as possible about these potential team members. Meet them. Have lunch with them. Have dinner with them. Go out with them. Learn about these people. You're going to be very, very intimate with them once the project starts, and you don't want any surprises. If someone potentially has a temper that's going to flare up, if I know that beforehand, then I will expect it to happen. But if that suddenly rears its head unexpectedly in the middle of a project, if suddenly I'm stuck with a diva, then it's a little too late in the game. It's a very awkward situation.

Do I have to fire this person? Do I have to sit and give him or her a lecture? But I should have known in the first place that this was going to happen. You have to try to build your team based on whether these people will be harmonious. Once it's gone wrong, I think you have the ultimate weapon: to fire a team member. And that's not a bad thing. That could be a very good thing, because if you get rid of the troublemaker, it'll strengthen the team. They'll know that you have the courage to do such a thing, and they probably didn't like the person anyway. And the next person who comes in is a Johnny-come-lately to the team, so he or she will be more obedient, do his or her best to fit in.

So, it's hard when you have different personalities, but if you're ready for the worst you can manage it better, a lot better.

Andrew: Firing people—that's not a topic a lot of people like to talk about.

Tony: You know, firing people is inevitable. You have to consider that can happen.

Andrew: I had a boss once who told me that the only way to get a team's respect is to make it clear that you're the boss. In his mind, I think that meant asserting his authority, and yelling at people until they did what he wanted.

Tony: You can't yell at people. If you yell at people, they don't hear the words, they hear the anger. You have to control your anger, and it's really not easy to do. You could be angry when you tell the truth, but you can't afford to show it. And you have to learn how to take a breath, go out of the room, come back in when you've calmed down a bit, and then tell the truth. You can't scream at people—it does not work. It just brings up memories of their parents or their teachers or bullies. You can't come off as that.

Andrew: But you do sometimes have a genuine difference of opinion, and a decision does have to get made. How do you handle that? What if you know things have to go one way, and you know that there's somebody you have to work with who just wants to see things go another way? What do you do if you know that it will just not work out if we keep going in that direction, but the people you're working with are just having a lot of trouble even seeing that?

Tony: That's delicate, because in my case, sometimes I have to back off every now and then, to realize that the artist's name is going to be bigger than mine in the credits. My team always contains a "star." Sometimes I have to change; I must admit if an idea comes that's not what I had in mind, I have to completely examine it and say, "This might not be a bad idea. I could be wrong."

So, I always keep that option open, but if I'm dead certain that this is going down the wrong way, I will have to remind everyone because every project begins with defining the goals. You have such meetings at the beginning of the project. And it should be down on paper, and charts, schedules, should be made up. We need to do this by that date, and to have a person go off the charts at some late stage in the project is really in breach of the contract which makes a team cohesive. I'm talking about the social contract of getting the job done. And if you had these things written down, you have to remind people that they've gone off the page.

That's very, very important, and that usually reels people back in, even the "star." But I do my best to see the difference between a frivolous idea and a brilliant idea. And of course, Andrew, if the idea is so amazing, you have to say, "I wish this idea came up at the beginning. We should spend some time on this." But you can't entertain frivolous ideas. It's really out of the question. You have to be very firm.

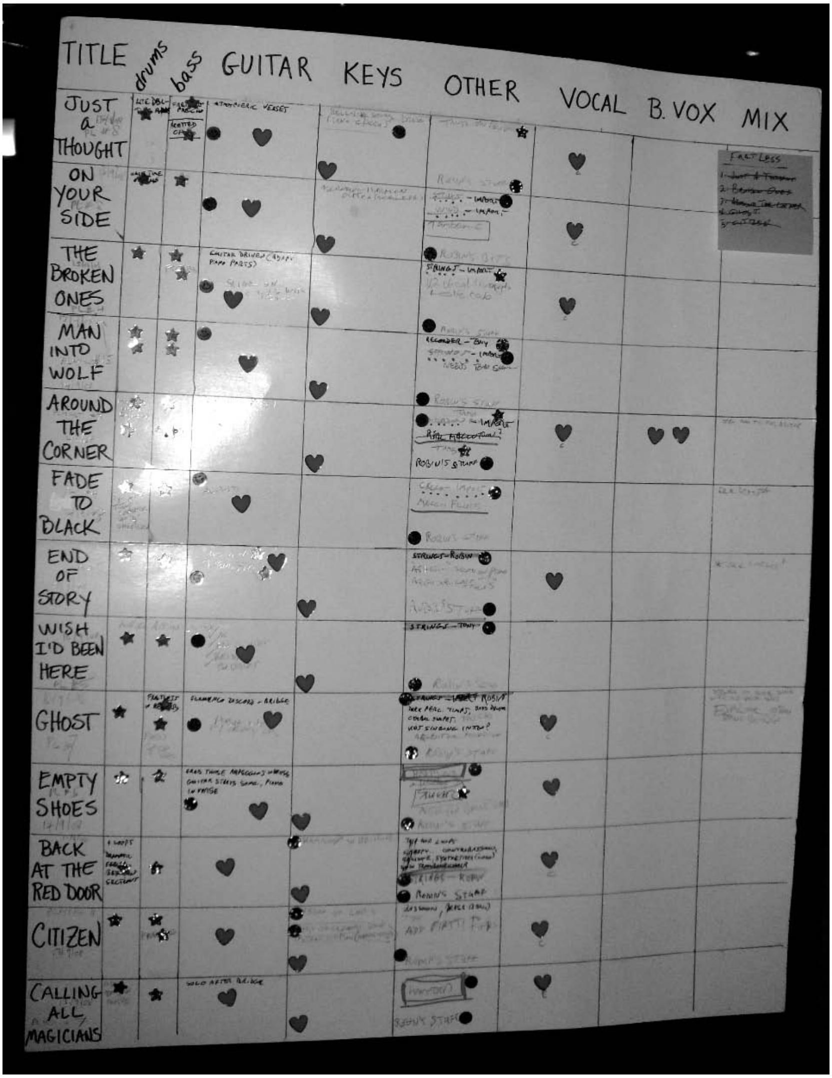
I'm a chart person. I have charts on the walls. In my studio right now, I have every song that we're recording up on a chart. I have horizontal columns for the songs, and vertical columns for what needs to be done to the songs. And every time an activity is done, it gets a mark—I put a random colored sticky star over the activity, which inadvertently turns into a grading system because the artist wants the gold stars. It gets kind of playful, too, because I encourage the use of the chart as an ideas section; anyone can jot down an idea. It makes everyone feel creative, involved, but mainly accountable, as it is always visible.

Everyone can see at any time at what stage we're at in the project, not just me. I think it's everyone's business to know how they fit into the whole. And my charts have been great for that. The other chart is a custom-made calendar from day one to the last day of the project, and it is numbered backwards, so the last day is one and the next-to-the-last day is two, so everyone can actually visually see what day we're in. "Oh, we have 45 days left. We have 42 days left. We have three days left." This is the bare visible truth about the deadline.

Now, I do this because I'm in a physical room, and I have these on the wall. But you could have an updated, interactive chart on your computer desktop that's emailed amongst your team every day, something that they open up several times during the day. We refer to our progress chart on the average of once an hour. If they're building a video game—"Oh, so Jimmy added the feet to that character, and Sue added a musical theme. OK, now I know what I have to do"—I really believe that a "public" progress chart holds everyone accountable.

Andrew: Is this something that you put up on the wall as a one-way communication to the team, or does everyone get involved in planning? Is it you talking to the team, or is it everyone talking to each other?

Tony: The progress chart starts out with all the ideas from the planning meetings. It is not a blank page. Everyone is involved with creating it. But as time goes by, I encourage everyone on the team to add new ideas that modify the original concept for the better.



TITLE	drums	bass	GUITAR	KEYS	OTHER	VOCAL	B. VOX	MIX
JUST a THOUGHT	★	★	ATMOSPHERIC VERSES ● ♥	● ♥	★	♥		
ON YOUR SIDE	★	★	● ♥	♥	ROBIN'S STUFF ● STRINGS – IMPORT ● TAMBOURINE	♥		
THE BROKEN ONES	★	★ ★	GUITAR DRIVEN (ADAPT PIANO PARTS) ● ♥	♥	● ROBIN'S BITS STRINGS – IMPORT ★	♥		
MAN INTO WOLF	★ ★	★ ★	● ♥	♥	● ROBIN'S STUFF RECORDER – TONY ● STRINGS – IMPORT ● NEEDS TONY SHAKER			
AROUND THE CORNER	★ ★			♥	● ROBIN'S STUFF ● IMPORT ★ ROBIN'S STUFF ●	♥	♥ ♥	
FADE TO BLACK	★	★	● ♥		IMPORT ● ● ROBIN'S STUFF			
END OF STORY	★		● ★ ♥	♥	STRINGS – ROBIN ● ROBIN'S STUFF ●	♥		
WISH I'D BEEN HERE	★	★	● ♥	♥	STRINGS – TONY ● ● ROBIN'S STUFF			
GHOST	★	★	FLAMENCO DISCORD – BRIDGE ● ♥		● ROBIN ● ROBIN'S STUFF	♥		
EMPTY SHOES	★	★	● ♥	♥	● ★ ● ROBIN'S STUFF	♥		
BACK AT THE RED DOOR	★	★	♥	● ♥	● STRINGS – ROBIN ● ROBIN'S STUFF	♥		
CITIZEN	★	★ ★	♥	● ● ♥	● ROBIN'S STUFF	♥		
CALLING ALL MAGICIANS	★	★	SOLO AFTER BRIDGE ♥	♥	● ROBIN'S STUFF ●	♥		

Tony uses charts like this to keep make sure everyone on the team can see what stage the album is at, at any time. How is this similar to Mike Cohn's task boards? How is it different?

Andrew: A lot of people would probably be surprised to hear that rock and roll album, this music, is made with charts and schedules and goals, writing down goals and building schedules. I think people would almost be relieved to find out that they're not the only ones who deal with those things, and that using a tool like that you can actually produce great art—and that it sometimes helps the creative process, rather than hurting it.

Tony: Well, anything that involves huge sums of money requires organization, and that includes rock and roll. There was a period when there was carte blanche, when Led Zeppelin's manager had an actual sculpture made out of cocaine in the form of a swan (and added to the recording budget). But those indulgences went a long way towards ruining the music business. Now the budgets have to be very streamlined and approved in the music business, because we're competing with video games and all kinds of other media. Pop music isn't raking in the kind of huge profits we saw two decades ago. Kids have more distractions now. Recordings have to be produced efficiently these days, especially rock and roll, I'm afraid.

But it's good. If people ask me, "We have enough money to put you in the studio with a group for seven days, what can you give us?" I'll tell them honestly, "I can produce four songs to a high level of quality." "Oh, but we want an album!" I can negotiate for more time, but if they insist on an album of, say, 12 songs, then I'll tell them that in that time we can record two songs to a high standard but the rest will be a jam, no sophistication. The premier album of the artist Jewel [*Pieces of You*] was just such a thing. They blew their small budget on the hit single, "Who Will Save Your Soul," and the rest of the album was taken mainly from her live performance in a club. You can't take a project on without being truthful about what restraints the budget will have on it. If you run out of money halfway through, there will be a drama and you will get a bad reputation.

Andrew: So, how did you learn this stuff? Clearly you've worked out a lot of systems, and know this isn't the sort of thing you can learn without making some mistakes. This is definitely the sort of thing where you have to have learned this from experience. I'm pretty sure that there aren't any books that teach you this stuff.

Tony: Well, I think it's pretty typical of people in their 20s to always want to come off as a cool person and as a nice guy, and that blocked me from taking complete charge. Eventually I saw that I wasted a lot of time and I allowed bad records to be made, because I was afraid to tell the truth.

Because eventually I grew tired of being, I guess *cowardly* is the word I want, and not having the courage to just call it as I see it. Once I did, though, I opened up the floodgates. I mean, I couldn't control it after that. I then had to learn diplomacy.

Andrew: But the charts—you basically came up with those yourself?

Tony: I innovated this chart in the '80s. I see a lot of other people using it now, the "public" chart. The people on my team were relying on my private notes for what was coming next and some were making private notes of their own. I thought, why don't we just put all the plans and ideas up on the wall for all to see? It worked great the very first time I did this. Now, when I work with the same artists again, they expect to see the chart!

I'm a bit fanatical about charts. I've got the backwards calendar that I already told you about. And there's one I have when I do a thing called "making a composite of a vocal." Say a singer sings a song seven times; I have a horizontal and vertical coordinate chart relating to virtually every word that the person sings, so I can cross-reference from, say, between eight vocals. After I pick the best-sung lines, words, even syllables, I am able to construct the "perfect" vocal. Anyone who wants to participate in the vocal comp can have the same chart and we compare our choices. I started doing the "comp chart" in the early '70s, but now loads of artists and producers use variations of this chart.

I find that these are interactive things that really bring the team together. They get really, really keen if they feel included in decisions. Still, I have veto power because I am ultimately responsible for bringing the project home. The charts I use are amazing team tools.

Andrew: That sounds really effective. It sounds like you've figured out a way to bring visibility into the project, and get your whole team involved; and it sounds like you need to do it in a way that balances out everyone's ego. Actually, it occurs to me that you need to deal with a lot of the same personality issues that you'll see on a programming team. When you're working with programmers, especially superstar programmers, sometimes the best people on your team are also some of the most socially awkward people on your team. They're not going to respect you unless you make it clear that you deserve their respect. Does that sound like some sort of situation you've had to deal with?

Tony: The human mind works in strange ways, and highly creative people generally have bad social skills because a lot of the brain is involved in being highly creative. They don't observe the effect that their impetuous outbursts have on other people, how they are alienating themselves from the rest. They don't see that. It is a social problem, but it's not very different from a young child who is very, very selfish. I'm not trying to make fun of creative people, but often, emotionally they haven't progressed much further than that in their social skills.

Often it becomes all about them, and like a lot of young kids, they need reassurance that you love them. That's really what this is about. It's a cry asking, "Do you love me?" That's what they want to hear you say. Now, of course, we're all adults, so you can't go around saying all the time, "I love you, Jim." You have to show it in other ways; you should make your appreciation quite overt. It might mean addressing Jim first in a meeting. You might have a group of them in the room; you say, "Let's discuss the next stage. I'd like to know how you all feel about it—Jim?" Jim might not have anything to say in that moment, but he'll notice he was the first to be consulted.

I probably sound arrogant, but as a father, I've learned to deal with children and divas. Every band I've worked with have sober, mature-thinking members and one or two diva-like personalities. There is a pecking order; it naturally forms as it does in all forms of social interaction. In a successful team we've all learned that we wouldn't be a team without our "star." Fortunately, a real smart "star" knows that they'd find it very hard to succeed, or even survive, without a great team.

I am a well-known record producer, but I can't afford the luxury of being a "star" myself. I have to be the soberest member of the team and lead my group of mavericks on into glorious victory.

Andrew: You know, I recognize some of that in myself. Early on in my career, when I was the top programmer on a small team, I acted like a prima donna. It was all about me. And I wonder if I would have gotten along better with the people around me if I'd said to myself, "You know what? I am a little bit overdemanding. I get impatient with people and I'm demanding, but I produce and they put up with me because I produce." What advice *could you give people who are like that early-career version of me, advice that might make dealing with others a little easier?*

Tony: I think they have to learn about the impact it has on the rest of the team, by being petulant and selfish. Even the most understanding people on the team will take only so much childish behavior.

And the programmer must realize that he might be a great programmer, but in a world of billions of people, there are other great programmers who will gladly fill his vacant seat after the mutiny. In this big resourceful world of talent, any team member could be replaced. Everyone has the responsibility to learn how to get along with each other, even the "stars."

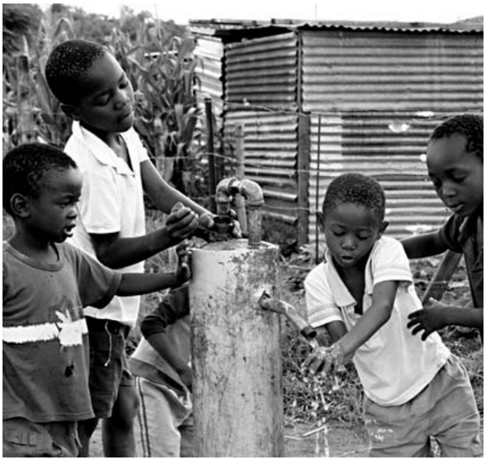
You're in competition with other A teams; there's always competition. You cannot afford to have internal squabbling. You cannot have infighting to the point where one individual is destroying the morale and the productivity of the team. Someone who's got a better attitude will replace you. Everyone's replaceable.

Andrew: Before we finish this off, do you have any advice for somebody who is a good musician, or somebody who's a good programmer, or somebody who's good at whatever job they do? Someone who sees the next step in their career is going towards what you do, going towards leading a team or managing a team or making sure work gets done well? What advice would you give them?

Tony: Well, nowadays there are so many distractions, and it's really, really hard to concentrate given what we are exposed to. My motto for musicians who make music on the computer, for instance, is that computer skills aren't really that difficult to attain—compared to real musical skills. A smart 8-year-old can work Garage Band; it's no big deal dragging a loop into a space and another loop into another space and all that. My motto is "Put down the mouse and pick up a guitar." We're in danger of losing the ability to concentrate on mastering something. Nowadays the temptation is so great to just get lost in skateboarding, video games, drinking—a lot of people start drinking too heavily, too early now. But the greatest musicians that have ever lived have spent a long time learning, practicing, and perfecting what they did.

And I think that even though you reached a certain level—like there's a lot of young musicians who can play 12 chords and they think, well, that's all I need. My advice is that the more you know and the more you study, you'll have a better chance at succeeding in life. That's it. So devote your downtime, devote it to learning more stuff. I'm still learning new stuff to make better productions.

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.



CONTENTS

Software Engineering

Beautiful Teams

"The good, the bad, the ugly, and of course, the beautiful. You'll read about teams who persevered no matter what, and how some famous organizations built teams. If you want to know how your team stacks up, or how to make it beautiful, read this book."

—Johanna Rothman, consultant and author

What's it like to work on a great software development team facing an impossible problem? How do you build an effective team? Can a group of people who don't get along still build good software? How does a team leader keep everyone on track when the stakes are high and the schedule is tight?

Beautiful Teams takes you behind the scenes with some of the most interesting teams in software engineering history. You'll learn from veteran team leaders' successes and failures, told through a series of engaging personal stories—and interviews—by leading programmers, architects, project managers, and thought leaders.

This book includes contributions from:

Tim O'Reilly
Scott Berkun
Mark Healey
Bill DiPierre
Andy Lester
Keoki Andrus
Tom Tarka
Auke Jilderda
Grady Booch
Jennifer Greene
Mike Cohn
Cory Doctorow
Neil Siegel
Trevor Field
James Grenning
Steve McConnell
Barry Boehm and Maria H. Penedo
Peter Glück
Karl E. Wiegers
Alex Martelli
Karl Fogel
Michael Collins
Karl Rehmer
Andrew Stellman
Ned Robinson
Scott Ambler
Johanna Rothman
Mark Denovich and Eric Renkey
Patricia Ensworth
Andy Oram
Tony Visconti

Andrew Stellman and **Jennifer Greene** are veteran software engineers and project managers. They've been writing bestselling books for O'Reilly since 2005, including *Applied Software Project Management*, *Head First PMP*, and *Head First C#*.

A portion of the proceeds from this book will be donated to PlayPumps International.

US $39.99 **CAN $39.99**
ISBN: 978-0-596-51802-8

53999

9 780596 518028

O'REILLY® www.oreilly.com