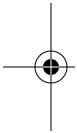


## CHAPTER SIX

# Keoki Andrus on Inspiring People



---

Keoki Andrus is an expert on improving the way teams work together. We first heard of him from a talk that he gave on performance and potential when he was head of operations at Intuit. He has some great ideas about how to take somewhat lofty concepts like “vision” and “potential” and turn them into practical, real-world techniques for improving the way teams run. Andrew sat down with Keoki to talk about this.

---



*Andrew: So how did you come to know about software teams?*

**Keoki:** Well, I've been in the software industry for 20 years. I started at Microsoft, and I worked on the first versions of Word. I worked with one of the project managers for Office, too, when we converted it from a bundle to a suite. After that, I worked as an executive for years, running product management at Novell in project management and strategy.

This became a family joke with my wife. We would always start the engagement with a discussion of "What should we be doing with our product?" And almost always, within a few days, it became a discussion of "How do I run my company? My people aren't happy. How do we solve these problems?" And it's always because of the way the organization runs, and the cohesiveness of the organization—those things have a huge impact on the success.

*Andrew: Where did you really cut your teeth on getting teams working?*

**Keoki:** I worked at Intuit for a while. I ran a company, and I was called in there. The Orem, Utah, site was a site that they had opened up six years previously, and it was really failing. When I talked to Intuit, they had kind of made up their minds that they were going to shut it down, they just didn't have the productivity. "Morale is low, and if we don't get it turned around we're gonna have to close it. So you're our last chance. Why don't you give it a try?" Well, I worked with that site and we turned it around from a low-morale, low-productivity kind of a scattered organization into one of the best performing group of engineers in the company. We doubled the size of the site. Some people got promoted, some people had to leave, but it ended up being pretty successful.

*Andrew: That's pretty impressive. How did you do it?*

**Keoki:** The first thing I did was that I went and just sat down with each individual in the site and tried to get a pulse from them—what they saw as some of the difficulties.

I had two objectives in doing that. The first objective was that I really did want to understand them, and what was going on from their perspective. The second thing I wanted to do was help them to understand that I truly cared. Because my doming hypothesis is that if people don't believe you care, they're not going to care about what you say.

I discovered pretty quickly that there was a lot of discontent. The team had several individuals who had very high potential and could really become high performers, but they were fractured. These people had a strong need for human connection, for good working relationships, but they had been peeled off one by one to work on projects as part of a remote team.

Now, some people can do that very well. But for people who need a day-to-day high degree of interpersonal connectivity with others that they're working with, that doesn't work so well. I noticed these people; they were leaders at the site and they were well respected, but they were grumpasaurus.

First I tried to tell them, “You’ve got to knock that off,” but they just couldn’t hear that message. Then I noticed that one of the managers at the site that I was managing was well respected by everybody. In fact, I would go beyond well respected. I would say he was loved by everybody.

And I grew to feel the same way, and I still think of him extremely highly. He’s one of those fantastic human beings of high integrity, but he was getting the crud kicked out of him by upper management because of the way his people were acting. So I pulled aside the leader among all the high-potential low performers and I said, “Here’s the deal. You can be a superstar, but you’re frustrated by some of the things going on.” He agreed. So I said, “Did you know that your actions are hurting your manager?”

That stunned him, because he loved this guy and he didn’t want to hurt him. So I sat down with him and I kind of went through his career possibilities and who he was.

By this point, he knew I really cared about him, because I hadn’t pistol-whipped him even though he’d been pretty open about some things he wasn’t happy with.

So I went to his remote managers, and I worked out a transition plan for him to get back on a local team. They were happy to get rid of him, because he was so unhappy. So he saw that coming. He could see that I was going to bat for him, and once we got to that point, I sat down with him and I went point by point through each of the strengths that I saw that he had.

I said, “You know, when you add all these up, you are destined to be a superstar, but this need to be a curmudgeon, to be disruptive in this negative way, is killing you. Could we just throttle that back a bit? I know it’s you, to be able to see these things that other people don’t see. You’re funny and you love to show that off. But could you just kind of keep that around here with the boys? Could we just not do that when we’re on the phone with people at corporate?” And he turned around!

Then there was the manager. I had to help him to start to see how he could manage his people. I’d been told by the vice president about that manager, “You’re probably going to have to let this guy go. He’s got a lot of potential, but right now he’s just dead wood, and we can’t figure out a way to get him to work successfully.”

So I went and met the guy, and I went through his life with him. Then I talked to the VP and said, “You know what, this guy is a superstar. He’s not marketing very well, but he’s a superstar. He’s just got a few adjustments to the way he interacts with others that he’s got to change.” So I talked to him just like I talked to the developer. Basically, I said, “Here’s why you’re great, here’s the stuff that’s holding you back. Let’s work together on not advertising that one thing so much, and we’ll shore up a couple of these areas and get you where you need to be.”

He started working on that. Six months later, the site was really moving forward.

***Andrew: It sounds like you started with a good team that had all of the right technical skills, but were being held back by that “soft” stuff that software people like us hate to***

*think about. And it sounds like you approached the problem by knocking down communication barriers—helping them talk to other people better, advertise their strong points, tone down their weaker points in front of outside people. Is that how you operate? Starting with communication issues?*

**Keoki:** I'm a one-on-one communicator, and I love to be with somebody face to face. I don't do as well on the phone. With that manager, it was actually kind of funny, because at first the vice president basically said, "Why haven't you fired him? This guy's a loser. You need to get a new manager in there." I had to tell him, "You know, I really think he's got some great qualities and great capabilities and we need to invest in bringing those out." That vice president actually told me, "You're the first person in six years to ever say anything positive about this guy to me. Are you sure?"

Later that year, I got my review, and my review was really good. But part of the review said, "It's so amazing how much you've managed to accomplish, seeing that you have to work with people like this guy and others who aren't very talented." I found that offensive, because I thought these guys were amazing.

So every time they did something good, I made sure it was genuinely good. It wasn't just marketing. These people were really genuinely good, but nobody knew. So I started making sure that proper credit was given to them when they did something good. And after awhile people started talking about that same manager and saying, "He's phenomenal."

And then the next year, a funny thing happened. I got my review—it cracks me up, because my review said, "Yes, your site was fantastic. But what do you expect, when you have an amazing manager like this running the place for you?" I sure wasn't going to say anything, right? Because that was what I was trying to do for him. But I thought that was hilarious.

*Andrew: So what did you learn from all of that? Is there some core principle that you took away from all of it?*

**Keoki:** One thing that I realized was that, going back to the beginning, these people had no vision. They just saw themselves as working at a job. Nobody sees their own greatness or their own potential. So I sprang some visions.

The first vision was that this would be the greatest engineering group in the company: that people would look to us as the vanguards, the trendsetters, the ones who were always on the edge of doing the greatest, most innovative things. We would be the group that you go to when you have a hard task.

Up until now, they were the street cleaners, the garbage men. "Somebody's got to code this. We'll throw it over to them." And I thought, "We've got to change that." And one of the things that I observed was that every engineer there really was talented. They were really talented, but the problem was that each one of them had their own message for software development.

They all had their own way of doing things, writing code. They all were good at it, and they all understood each other's methods well enough to get along with each other and adjust on the fly. But if anybody new came into the group, they couldn't comprehend it. So the organization had no ability whatsoever to scale.

If you were an old-timer, great. If not, you can't swim with these guys. And from an outsider's perspective, if you're working with them you can't even talk the same language as them, even though you're a software guy. So I brought in a group on software process, and we implemented a very hardcore process, which everybody whined at me about.

**Andrew:** *I've been there, gone down the same process road. You must have run into some resistance from the team, because they didn't want to change the way they built the software.*

**Keoki:** They were all saying things like "This is terrible, it's so hard." I said, "You know what, guys? We're about scaling. We are about creating an organization that delivers super-high quality on time, and you guys are capable of doing it. I know this is miserable." I knew that these guys would want to monkey around, and that they'd rebel. I also knew that no process fits everything, and that there's always some part of a process that's stupid.

And I have a philosophy that I live by. Everybody that works with me knows this; it's on the wall: "If stupid enters the room, you have a moral duty to shoot it, no matter who's escorting it."

**Andrew:** *Did people take that idea seriously? Even if it was your idea that they thought was stupid?*

**Keoki:** People who work with me know that. If I'm escorting it, shoot it. I don't have any problem with that whatsoever. I will laugh if you point out how dumb I am. There's no ego with that.

What I wanted to do was get them to just give the new way of doing things a try, because everybody had this resistance of change. I was hard-nosed at the beginning: "We will do everything in the process, no matter how stupid it seems." The response was "Ugh." But at this point, they trusted me some, because the site was now getting credibility, good things were happening, they were getting treated well, they were enjoying their work, and they were sitting there going, "Well, Keoki's done these other things. Maybe this is OK, too, but it seems stupid."

So they trusted me enough to do it. I said, "Guys, just hang on. Run with it for a while, and then when we find the stupid, and when we're sure that it's stupid and don't just think that it's going to be stupid, we'll change it." I got them to run with it for a few months. There were a couple parts of it where they said, "I don't get it" or "I know MIT came up with this, but it's dumb."

I'd say, "Well, let's talk about why it's dumb. Is it dumb because you just don't like it, or is it dumb because it doesn't fit our circumstances? If you can explain to me why it doesn't

fit our system, our particular circumstances, then we'll change it." We implemented a few changes, which actually helped the organization.

We got better at the process. At the same time, they realized I live by my philosophy; that I wasn't going to make them do stupid things just because that was the way it is. We got through that phase, and then we started delivering.

Compared to everybody, the other development organization, we were on time. And our quality was insanely high; it was like a couple of quantum levels above everybody else's.

**Andrew:** *That's a pretty impressive result. But it strikes me that you didn't do anything magical—a lot of it sounds like common sense. You mentioned that you had sprung several visions. What did you do next?*

**Keoki:** The next part of the vision was "Well, we have to be the greatest." Well, what do the great engineers do? They innovate. So we launched a program of trying to really think outside the box, to figure out how do we innovate.

We started creating patents. In the first year that we started this program, where we really put emphasis on innovating to create new intellectual property, we only represented one-half of one percent of all the engineers at the company, and we innovated like seven or eight percent—we had seven or eight percent of all the patents.

In the next year, we had 16 or 17 percent of all the patents in the company.

**Andrew:** *That's a huge jump. What did you do? Mandatory weekly brainstorming sessions?*

**Keoki:** One way I accomplished that was I did very little screening on people-submitted ideas. I wanted the ideas, I wanted them to flow, not to criticize. Then, later, we'd let the screening people at the company screen out some ideas, and say these are the ones we're interested in.

And it was funny, because I got an email once from legal, and they said, "You know, we need to talk about what appropriate patent idea submissions are and which ones are or aren't appropriate." And then somebody got a hold of them and said, "Shut up! He's the only guy doing what we actually want him to do." And I got a nice, polite little email: "Never mind, we don't need to talk about that."

**Andrew:** *I still want to know how you got that level of innovation, because sometimes innovation seems, well, almost mysterious to me. Where do ideas come from? I know we want to talk about teams, but it seems to me that those two things aren't unrelated. How do you get a team of people to come up with good ideas? How do you improve, how do you help people innovate in groups?*

**Keoki:** There's a general process of leadership which is key to all of us. The first thing is to establish a vision that everybody believes in passionately, and that's a process of inclusion and ownership. Then the second thing is the how. If people own the vision, they will internally create passion that creates the innovative juices.

When we discuss these things within the team, I don't come up with all my own ideas. I socialize them into the group. We negotiate until we get to the point where everybody's passionate. I brought up this thing with the patents with them, and they got turned on about it, because it was part of our vision for being great and putting our site on the map.

Once that happened, then whenever we'd be in meetings, somebody would have an idea, someone would always ask the question, "Is that patentable?" And people would start talking about it: "Wait a minute; that could be patentable." After awhile, I would go around the office—one guy had on his whiteboard a big thing just listing all the ideas that were active in the group.

It became part of the culture to figure out how we could innovate. People got excited because there was always a question: "Is that patentable?" and you've got to follow up on that. It was all part of the vision, how the team saw themselves. If you set a vision, you've got to follow up on what's important, and when people see it, if they have that personal passion, they really will create spontaneously.

The other thing is that I had to create some urgency. For example, there was this one engineer, a really brilliant database guy, and he never wanted to overstate what he can do. Like many engineers, he said, "I don't want to promise what I can't deliver." It took me over a year for these guys to realize that I would not chew them up for being bold, for learning something new that changed their schedule.

Man, be bold. That's a rule. If we learn something new, well, we're smart guys. We know that we learned something new, and that changes what we thought. Only imbeciles chew people out for not having known everything that they would yet learn.

Yet, engineers always get punished, because everybody wants a schedule from them. Well, I don't play that game. These people started to get that. This engineer, he never wanted to promise what he couldn't deliver. When I realized that, I figured, "I got his number."

He'd come into my office and he would say, "You know, we can't solve this problem. It can't be done." And I would look at him and I would say, "Jeff, you are one of the most brilliant engineers I've ever known in my life. I just have a real hard time hearing the words 'it can't be done' coming out of the mouth of one of the most brilliant engineers of all time.

"Tell you what, Jeff, you go back to your cube, and you figure it out, and at three o'clock, you come back and tell me what the answer is." He'd just look at me and say, "I just told you it's impossible."

"I know, for any normal engineer, it would be impossible, but you're not a normal engineer. You are one of the best." Now, you can't do this sort of thing if you're not sincere. If you don't have a close, intimate relationship with people, where they know you really mean it, then they can tell it's all bull.

**Andrew:** *But you had that sort of relationship with this engineer, so he trusted you and believed you when you told him that, right?*

**Keoki:** Right. He knew I respected him. I've done this so many times with him, and it's only him, right, because you've got to know every person individually. He'd say, "Oh, I hate it when you do this to me."

Every single time, within two hours either he'd burst into my office going, "I got it!" or I'd walk by his cubicle and he'd say, "Oh, come here, come here, come here, come here!" Because he had that brilliance, and I just had to put him in a position where he was challenged, where he felt personally motivated to tap into it.

It worked because he knew I wouldn't chew him out. Because I'd say, "If you can't figure it out by that time, then nobody can, but I think you can." And he never wanted to let me down.

**Andrew:** *It sounds like you started with a real solid group of engineers. You had some top-level talent, a team that had the potential to do really good work that was just held back by personality, communication, and vision problems. And it sounds like one important way you got it all to work was that you figured out what makes each person tick, and you gave him exactly that.*

**Keoki:** I had another engineer—I hope to get to work with him again. He's a god. He finishes his work ahead of schedule with great quality, then goes around the rest of the team and helps them solve their problems, and then he goes outside the team and fixes everybody else's problems.

This guy, his thing is play. He will not turn on unless he's in the state of play. He loved to play foosball and other games. Now, some people say, "Well, you're not working, you're not doing a good job." I say "bull". You have to tap into what turns on the creativity for the person so that they are operating in a state of passion.

Man, he loved to play foosball. And we had a thing in the group. There's two guys on each side on foosball. So if someone needed a game, they'd walk out and you yell, "Three," indicating that number four's already been taken. Somebody else yells, "Two," meaning we need two more, and then somebody else yells, "One," and then some gal says, "I'm in!" and we've got all four now.

You'd go play a game, and then you go back to your desk. Well, some people would say that's stupid. "You're encouraging people not to work!" But I detected that for many people, if you took 10 minutes and went and played a game, they would go back to their desk with this kind of creative, passionate energy from the competition and solve problems.

So I sat down with Jason, that brilliant engineer, one day, and I said, "Jason, tell me about how your mind works. You play games all day long, but you're the most productive engineer I've ever encountered in my career. What is the connection between play and performing at this super-high level?"



And he said, "It's just what I need to clear my mind. I get these problems, and if I play for a little bit, I go back and I can solve them." And I said, "Jason, here's the deal. I don't know if I'll say this for everybody, but if you ever feel the need to play, I want you to play, because if that's what it takes to get this kind of amazing performance, you should play as much as you need to."

And he never let me down. So that was part of your answer to how do you get that creativity going. You have to find what turns on passion for each individual and then bring them in as a leader into that state of play.

*Andrew: You keep using, it really strikes me, you keep using words like respect, passion, creativity, even love. Plenty of people talk about respect and creativity when they talk about motivation, but I rarely hear people talk about motivation in the language of emotion. But it sounds like you use respect and love as management tools, the way engineers use engineering terms. They're almost like jargon—in the positive sense of the word jargon. When you use the word love, you're not using it in the wishy-washy Hallmark sort of way. It really has a technical meaning, and it turns into an important tool to help a team gel. How did you get to that point?*

**Keoki:** So first of all, I discovered early in my career, I'm good at observing. Engineers like to think of themselves more like Spock than anybody else. "I am logical. That is all I do."

So I started observing how engineers would respond when somebody criticized their code. And sometimes I'd actually think, "I got a 2-year-old here. This person is incredibly emotional." The more I watched engineers, the more I realized they were as emotional—if not more emotional—than everybody else, and that emotion was a big part of their craft. They just got really good at kind of structuring the way they communicated.

I realized that they need an emotional outlet and connections, just like everybody else. So that was the first thing I figured out.

*Andrew: I'm wondering if this ties back to something else you mentioned that I wanted to ask more about: when you're coming up with ideas, having an overarching vision. You seem to use vision to improve performance in a real, measurable way to improve engineering, in a way that people who would otherwise not be inclined to use that term can respect.*

*And it strikes me that the word vision is another one of those words that hard-core engineers and developers might not use. It probably sounds like business-speak, management-ese jargon. Might not—I know I haven't gotten it in the past.*

*Personally, it took me a long time to get any traction with the concept of a vision for a project, or for a team. I know that vision has a lot to do with figuring out who you're building the project for, with really meeting their needs, and with understanding what problem it is that you're solving. But just hearing you talk about it, I'm wondering if maybe there's a more visceral definition for vision. Can you help me understand the idea*

*behind vision, something that might help somebody who's looking to really come up with better ideas, or to improve the way their projects are run?*

**Keoki:** Sure, I can address that. Vision is typically botched.

And the way it's botched is like this: "Our vision is to become the number one provider of telecommunications equipment in the market." Or, "Our vision is that we're not building buggy whips anymore, we're building horse motivation tools." We come up with visions that are sterile.

If you want people to act with power, you have to tap into their emotions. Emotion is where creativity comes from. It's the source of innovation, and of long working hours, and all the other things that you want out of people. It doesn't come from holding a gun to their heads. And boring doesn't work. So how do you get a vision that evokes that?

If the people on the team can't see it in their minds, it's not a good vision. So you have to be a little bit of a televangelist in the way you paint the picture of the vision.

Let me give you an example of a vision I had. And I think you should have vision all the time. So my sons have been in show choirs. Do you know what show choirs are?

**Andrew:** No.

**Keoki:** Show choirs are kind of like Broadway for high school. And these choirs sing, and at the same time, they dance—they do almost drill team at the same time. It's quite amazing. My second son is a national champion, and my first son was picked as the best male performer. We were going to go on this trip to Branson, Missouri, with his group.

I live in Utah, and it's a 30-hour bus ride. We could do a plane, but people wanted to go as cheap as we could. And they went to the adults and they said, "We need some parents to volunteer to sit on the bus with these teenagers for 30 hours each direction." And parents signed up, but everybody had this attitude of like, "Well, we have to do it. I'll be swimming in a stew of fatigue, estrogen, and testosterone for 30 hours. Oh, it's gonna be awful."

And I thought about this and I said, "I'm going on this thing. What am I going to do? Am I going to just sit for 30 hours each direction and then just be miserable that I'm stuck with a bunch of teenagers? No, I need a vision." OK, well, what kind of vision could go along with this?

And I started thinking about it, and I said, "Wait a minute, these kids are going out and doing something that's one of the great opportunities in their young lives: to discover their potential. So wait a minute. I'm a guy who's totally into leadership and helping people discover their potential. Is there anything I could do?"

I thought, these kids don't know how to do any of that. What if I made it my goal to make this one of the crowning moments of at least some of their lives? It could be that thing that they looked back upon and think, "That was the time that I really felt my potential. That I really touched greatness for a moment."

That changed my job for the trip: it wasn't to baby sit these kids for 30 hours each direction and while they're at the hotel. My job was to find every moment I can to show them a little way to tap into their own greatness. That was what the vision was for.

And as a result, just having that vision and looking for it every moment of that trip—well, first of all the bus rides were a breeze for me. But second of all, I was able to take leaders aside, one by one, when I could see that they were at a moment where they didn't know what to do, and show them a little bit about greatness. And as a result, they actually accomplished amazing things, and it was a great experience in their lives. I'm still friends with those same teenagers.

I shared that with you just to kind of give you the picture of how simple a vision can be, and how mundane it can be. What I mean is, it can be associated with something so mundane, but I hope that the way I told it to you gave you that visceral feeling, where you thought, "Wow, I wish I was on that bus with those kids." And that's what a great vision does. It makes you feel like "I want to be part of that." Even something as boring as being on a bus for 30 hours with adolescents.

*Andrew: I've definitely heard about projects and thought, "Man, I wish I could have been on that team." It's probably because they had a good vision. I saw an interview once with some video game developers from one of my favorite video games, and I thought that.*

*You know, one of the things that I've written about, and something that I've read many times from other people, is when you start on a project, you want to get it started out right. You want to avoid problems down the road. And everyone—including Jenny and me, in our own books—says that one thing you really need to do is lay out the vision in the beginning of the project. But I never really had a good litmus test to be able to tell you that a particular vision is good. And maybe you've just given us the answer. A good litmus test of whether or not you have a good vision may be that just hearing it makes you want to be a part of the project.*

**Keoki:** Exactly. And the same thing works on teams when you're recruiting. I always get the people I recruit, because when I recruit them, I don't do the boring song of "Here's what your duties will be." On my team, everybody is on their way to somewhere in their career, and it's unique for each person. I talk to them about that.

My goal as a leader is to find where that right place is for them, that they agree with, and craft a vision of where they're going. Then my job is to help them get there in the context of them doing good work for the team. I'll teach them leadership. I'll teach them anything that they want to learn so that they can go where they want to be.

I'll give my recruits examples of how I do that with people, and then I say, "Now go talk to the members of my team. And you can ask them anything you want. Ask any member of my team." I'll let them talk to people already on my team, and they'll say, "Yeah, he's completely dedicated not to just my career, but to me as a person."

They always come back and say, "I've never encountered a job like that, and I would give anything to be in a team like that." And so, I always win.

*Andrew: Wow, I've done a lot of hiring over the years, and I've never done anything like that. Let's say I want that for my team, I want that for my company. I want my teams to work that way. I can see what it will look like in the end, but how do I get there? What's my first step in getting from where I am now? You told a story of how you started with a team that had a lot of potential, and you turned them around. Let's say I've got a team, and I know they have a lot of potential. How do I turn them around? How do I get them from point A to point B?*

**Keoki:** OK, so one thing is you can call me up, and I'll come out and help them. I do this for a living, you know.

*Andrew: (Laughter.)*

**Keoki:** But I do love doing this. The process of developing skills is a process of first recognizing that you need the skill, then learning the skill and then processing it. Then, to actually get to a master, you need the tutorage of a master.

And you can add that as much as you can in the book, but you really do need coaching to become great at something. Now, having said that, here's the good news. People are hungry to actually be cared about. Think about that for a minute. You want to be understood, right? Everybody does. Everybody wants to feel important. It's universal.

Now, if you just try to do this by turning to your team and saying, "Oh, yes, I will do this now. I don't feel anything, but I'm going to do it," then forget it, you're just a pointy-haired boss.

But if you sincerely just reach out and start with trying to get to know your people as human beings, with the mindset of trying to understand who they are and what's important to them, trying to be a facilitator of their vision or helping them to create a vision, people will respond exponentially to being treated with care and compassion, because they've been dying for it.

They're starving to death. Imagine the old cliché of the man who's dying of thirst in the desert. How much will he pay for a drink of water? Well, people are starving to death in virtually every job, looking to actually be treated like a human being.

I've never encountered anybody, except people with borderline personality disorder, who don't want to be understood. Some people will say they don't, but just watch their behavior and you'll realize they do.

And that's what I have found as a leader that I often do with people is just sit down and go, "Oh, you know what, you missed it. Let's talk about how to succeed with it." And I know that that wasn't the greatest answer, but you can't magically jump to it, but you can have great results just by trying.

*Andrew: So how does that work in practice? If I were bringing you on board, say, to help me improve the way I ran my own team? Right now our team is me and Jenny—we're working on a book called Beautiful Teams. We're leading a team of about three dozen people who are experts at building software, and we're all working toward this one goal*

*of writing a book that's both informative and interesting. What's the first thing you'd do for us?*

**Keoki:** I'd ask you to tell me more about yourself, you and Jenny. I feel like I hardly know much about you. So tell me—right now.

**Andrew:** *Right now?*

**Keoki:** Sure. What do you do? What's your vision? Who are you?

**Andrew:** *OK. Well, basically, Jenny and I started writing books for O'Reilly a few years ago. We had a lot of fun writing two books in their Head First series, writing one on project management and one on learning C#.*

*Before that, we wrote our book on software engineering and project management. To be honest, it's really a book on quality, on making software better and software projects run better—we kind of realized at the time that our goal was to make the world a little bit better by helping people build better software. And that's really what we do: we help people build better software. It's why we write books and speak at conferences, and it's why we've done so much consulting over the years.*

*That's actually why we've spent so much of our careers targeting project managers. One thing we figured out early on was that project managers are an underused, under-respected resource that can really help build better software. If a project manager takes the time to understand how software is built, he can do amazing things for his team. So I guess that's our vision: we want to give people the tools they need to build better software, or maybe to help them solve the problems that are keeping them from building better software.*

*And other than that, I've spent a lot of my life first building software and then leading teams that build software. Jenny spent a lot of her life first testing software and then leading and managing teams that build software. I don't use the word brilliant all that often, but she's a brilliant quality engineer. Before I met her, I felt that software testing was a kind of "black art," something you're kind of born with, or maybe just feel your way through. It wasn't until I worked with Jenny, back in the '90s, that I realized that software testing is a real engineering discipline in its own right. You can understand it, you can make it transparent, and you can get a team to do it. And she does it brilliantly. I'd deliver software to her that I thought was done, and she'd find so many bugs that I couldn't believe I was willing to let it out the door. But more than that, working together we figured out how to keep a lot of those bugs from getting into the software in the first place. Doing that really made me really realize the art of quality, of building better software, as well as the science.*

**Keoki:** There's a certain way that that type of QA person thinks and what motivates them, and if you got it, you got it. I was a QA engineer at Microsoft when I started out, and I've known some of the people I think like you're describing.

**Andrew:** *Jenny was the first real QA engineer I met who really understood quality, really got it on a deep level. Not just on a "let's make this software not crash" level, but really understanding it's about how software works, how software meets the users' needs. It's*

*about software conforming to requirements—getting back to the real definition of quality. And that’s what motivates her—it’s what motivates me, too.*

*Quality’s also about process—but process is more than just getting people to do things in a certain, repeatable way. It’s more than that. It’s about changing the way an organization breathes and grows, it’s about changing the way people actually act and, hopefully, how they think about their jobs.*

*We were working together for a few years at a small financial software company and, through a lot of trial and error, we were lucky enough to be working at a company that was small enough and inexperienced enough to let two relatively young, inexperienced people change, completely change, the way they build software.*

*And we got some really good results. We also made a lot of mistakes along the way. And it was really good for both of us. I kind of feel really privileged to have been able to make such mistakes. And we dedicated a lot of our career to understanding mistakes, understanding why projects fail.*

**Keoki:** I actually have my top 10 list of how you create passion, and one of the biggest ones is celebrating mistakes.

The one talent that virtually every person in the world feels like they have that is unique to them and they feel a moral obligation to share is the talent of criticism. “I am the one who can find what’s wrong with this, and since I can, I must tell you what’s bad. I feel obligated.” And we’re all conditioned: we all hate it, right, but we all are good at it.

But the thing is, have you ever been in a situation where you’ve been, say, in a meeting, and something bad has happened? Now, that may mean anything. Say we shipped the software, and there’s a crashing bug in it, and it’s terrible.

We’re sitting around talking about it, and somebody says, “We need to have a postmortem on this so we can figure out what happened.” And then these words come out: “So that it never happens again.” Have you ever heard that?

**Andrew:** *Yes, of course. Most of us have.*

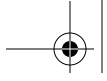
**Keoki:** OK, let me ask you this question then. Can any success happen in the absence of risk?

**Andrew:** *That’s a good question. Probably not, I suppose.*

**Keoki:** You will get a zero rate of return if you have zero risk, because there’s no risk, so why should you be compensated? No success ever happened without the presence of risk. But what is risk? Risk is the chance that something bad will happen. Do you buy that?

**Andrew:** *Absolutely.*

**Keoki:** OK, so if no success can happen without risk, and risk means that something bad can happen, then if you set out to make an environment where risk goes to zero, what else are you driving to zero? The probability of success.



*Andrew: I never thought about it that way, but it does follow.*

**Keoki:** So it isn't bad to say, "You know, we may not have assessed our risk correctly, and we may have exercised poor judgment in something we did. Let's see if we can learn what we might have screwed up on so that we can be more informed in the future."

But the next thing that happens is that once you figure out what went wrong, you go after the person who made the mistake. And everybody else sees it; the head rolls down the hallway and gets put on the city gates. Everybody learns, "Don't make a mistake." And it's in the very act of creating that stress, that stress of not making a mistake, that you create the situation where nobody will take a risk.

Instead, what you must do to create that passion environment is when a mistake is made, you say, "OK, what was the mistake? OK, well, we did that. Good job. Now we know not to do that. Now we are going to be more sensitive to that in the future." You can even give the person a reward.

Now what happens? It creates an environment where people don't feel afraid to try something that could be risky. You want to encourage people. Now, if people take risks, and a reasonable person in possession of the facts can see that a decision was made that was clearly not thoughtful, then sure, they should get it right in the teeth for that. But nobody thinks that's unreasonable.

But if they were making a reasonable decision that just turned out bad, that just means they took a risk, which means there's a probability of something bad happening. And guess what? Sometimes that's the way it comes out. And if you don't freak out, but rather embrace it, then people feel at ease to take chances. And when they take chances, they innovate, and that's where greatness happens.

And that's why often, companies that are successful start out taking a big chance. They get their success, and they spend the rest of their corporate lives trying to drive success to zero, because they're afraid of losing what they have. And the very act of trying to not lose guarantees failure.

