CHAPTER FOUR

Building the Perfect Team

Bill DiPierre

I CLEARLY REMEMBER THINKING, "MAN, I AM OUT OF THE LOOP." I SAT IN A COMPUTECH CONFERENCE room with a newly assembled team of high-powered developers. The task at hand was to port our flagship financial software from Microsoft QuickBasic to the promising new Microsoft product, Visual Basic 3. As a resident VB guru, I would bring technical expertise and hard-earned experience to a project that would determine the future of our firm.

It was hard to believe that only three weeks before, I had finished my final day of work at the large commercial bank where I cut my programming teeth. Only five weeks before, I'd met Jack, my current boss, for the first time. I'd emerged from that interview feeling a strong connection with Jack, but also insecure about how well I'd impressed him technically. And of course, perhaps hardest to believe, it was only six weeks ago that I had plunked down $11.95 at Barnes & Noble for a brand-new copy of *Visual Basic for Dummies*.

In the conference room that day, Geoff and Mark, the two dev leads for our team, were debating some fundamental architecture questions. Mark laid out his argument, saying, "Look, I know it will be a lot of extra work upfront, but I don't see any way that we can get the UI experience we want if we don't write our own user controls."

Geoff responded, “You’re crazy, Mark. Do you realize how much effort that takes? I’ve been working with C++ a lot longer than anyone here, and I’m telling you we are biting off more than we can chew.”

Mark was not deterred. “I just don’t see how we have a choice. The control suite from Microsoft does not do what the design calls for.”

“Then the design needs to change.”

“That’s not what we’re about. We make the technology work to give clients what they want. We don’t just regurgitate the latest technology from Microsoft and tell clients to live with it.”

“Well then, why don’t we give them a teleporter? Since we’re not limited by technology, we might as well help them with their commute.”

Sensing that the constructive portion of the debate had come to a close, Jack stepped in, saying, “All right, all right. Before we commit to building a teleporter, let’s back up a step. We need more information. You guys are my experts, and I need your expertise right now. Geoff, can you put together an estimate of how long it will take to do what Mark is proposing?”

Geoff rolled his eyes and let out a big sigh. “OK, but I can tell you right now it’s gonna blow our timeline out of the water.”

Jack, unflappable, said, “That’s all right. The dates are my problem, let me worry about them. You’re our C++ man, so I need you to put on that hat and tell me what you see.”

Geoff nodded. “OK.”

Turning to Mark, Jack continued. “Mark, please give me a list of all the functionality we can get off-the-shelf without doing any custom work.” Sensing Mark’s reaction, Jack pressed on. “In addition to dates, I’m also the one to worry about the design requirements. You’re going to be implementing these babies in Computech, so let me know how far you can get with the Microsoft stuff. Don’t worry about what design asked for, just focus on what you can do with the off-the-shelf tools.” Then, with an almost imperceptible wink, he added, “And remember: I’ve seen your code, so I know you can do a hell of a lot.”

Mark, unable to resist a half smile at the compliment, said, “OK, Jack, I’ll see what I can do.”

Most of us remember the 1990s as a heyday for developers, and it was. The tech bubble was in full bloom, and venture capitalists were everywhere, looking for people to give their money to. And those people were programmers. We’ve all heard the stories of the rock-climbing walls and $10,000 coffee tables. The things that made it all go were the programmers. They were in demand, they could name their price—even if their price was a rock-climbing wall and an extremely expensive coffee table.

In the Computech conference room that day, the rise and fall of the tech bubble was still in the future. It was the early '90s, there were indications that software was a pretty good way to make a living, but none of us had any idea just how good. The demand for developers would quickly outstrip the supply, and turnover industrywide would reach record levels. Ten years later, I would look around the Computech conference table and see the same faces I had seen before the frenzy began. That is a testament to team solidarity. Not one of us had been lured away by rock walls or coffee tables. Actually, that's not quite true; Doug did leave, but he doesn't really count. He went on tour with the rock band he managed. I'll get to that later.

As a business, Computech was well positioned to offer a competitive product to a niche of consumers eager to get it and willing to pay for it. That's a great position to be in. As a technology, Computech was running up against serious limitations. It was written with QuickBasic and was designed to run on a single machine featuring DOS. As computer networking began to grow in the financial business sector, more and more clients were migrating their infrastructures to Windows. They wanted a Windows version of Computech. And we wanted to give it to them. Expanded network support, more robust memory management, and better development tools were just a few of the reasons Computech was determined to make this transition. Not to mention a whole new world of interface possibilities. Hey, we could use a mouse!

So there we sat. We had a reliable but limited DOS application. We had consumer demand for a Windows version. And we had a team of seven guys who needed to take the former and turn it into the latter. What a great project. It was big, it was complex, it was vital, and it was time-limited. Our need to retain a hard-won customer base meant our margin for error was small. The upside at the time seemed unlimited; this project would carry Computech into the future. It was a classic software team project.

Of course, if our two dev leads could not get along, we were in for a bumpy ride. After the meeting, my top priority, unofficially, was to try to get a handle on how this power struggle would resolve. In an attempt to get some impartial insight, I went to someone who wasn't directly involved.

"Doug, what's up?"

"Yo, Bill, how's the first month treating you?"

"So far so good, man. Meetings like that keep it interesting."

Doug seemed kind of surprised. "Yeah, I guess so."

Well, that was a non-starter. I tried a more direct approach. "So what do you think, we gonna write all this stuff from scratch in C++?"

"Huh? Oh yeah, I don't know. Jack'll figure it out. Probably do some of them and see how it goes."

Obviously, the teleporter debate did not leave much of an impression on Doug. I pressed on, saying, “Really, why do you say that?”

Doug shrugged. “I don’t know, just seems like what he’ll do. He’s pretty mellow and he usually will try something out and see what happens.”

All well and good, but I wasn’t getting much sense of how the office politics factored in. Doug, for his part, seemed oblivious to them. In retrospect, perhaps Doug was not the ideal person to shed light on these dynamics. Doug was not a programmer by choice; he was a programmer by necessity. Programming paid the bills during the gestation of his real career: rock-and-roll entrepreneur.

Doug had been writing code since his teens. He started using the MIDI package on his Macintosh Plus (featuring, of course, the fully expanded 4 MB of RAM) to create records in his parents’ garage. By age 27, he’d traded in the Mac for a Yamaha mixing board. His records were now circulated to non-family members. He also served as manager for an actual band. Doug spent most of his time consumed with music and was preposterously inept at downplaying this detail during job interviews. What most potential employers saw as a liability, Jack saw as an asset. Here was a kid who could bring strong technical skills without rattling the hierarchy. Doug wasn’t looking to move up the company ladder or leave his mark on the software world. Doug was looking to pay his rent until his band hit it big.

That was great for our team; Doug produced a lot of code. It wasn’t helping me get in the loop, though. A couple of days later, I was working with Geoff. I’d approached him for assistance understanding some old C++ code that needed to be ported. He was a big help. The code contained several optimizations that I didn’t understand, but the rationale behind them was clear to Geoff. I was impressed, and told him so. “Geoff, that’s pretty impressive. It would have taken me the better part of a week to figure out what we just covered in an hour.”

He was almost gracious, responding, “Yeah, well, I’ve been writing C and C++ code for a long time. Some things start to become second nature.”

“Yeah, I suppose. Anyway, I appreciate the help.”

“No problem. Let me know if you need more.”

That was encouraging—sincerity instead of sarcasm. I decided to press on. “Thanks, man, I will. Hey, so what’s up with our UI controls? Have you spoken to Jack?”

“No, I haven’t had a chance, but I will. It’s a big mistake.”

“Really? Seems to me like Jack is taking a reasonable approach, looking at his options.”

Geoff shrugged, unimpressed. “Jack’s problem is that he thinks Mark can do anything and it just isn’t true.”

“What do you mean?”

"I mean Mark is the main UI architect and that's fine. But he doesn't have half the C experience I do, so when it comes to a project like this he's in over his head. I don't think Jack sees that."

"Yeah, maybe. Maybe Jack just thinks it's worth the risk."

Geoff was moving from unimpressed to irritated. He fidgeted in his chair, saying, "Look, you're a VB guy and you just said you couldn't figure out this stuff without me. It's freakin' complicated. I mean, I can do it, but I've been at it a long time. It's crazy to add this onto the pile of work we already have when there's a solution out there that will do 99% of what we need."

I never said I couldn't do it without him, I only said it would have taken me longer. I let that slide. I really wanted to understand Geoff's beef with Mark, so I decided to avoid the personal slight and stick with statistics. Trying to maintain a tone of impartiality, I said, "Well, yeah, but the point is it won't do 99%. I think we're talking more like 59%."

"Whatever; it's still gonna be a big improvement over what clients have now."

I went with the loyal soldier argument. "Well, Jack said he'd take care of it, so I suppose we should let him worry about that."

Geoff found this unpersuasive. "Jack can worry about it if he wants to, but if he thinks Mark can just hop in and start cranking out C++ code, he's making a big mistake."

Somebody was getting cranky. I stuck with what I thought was the logical response. "I'm not sure, Geoff. From what I've seen, Mark seems like a pretty sharp developer."

"Sure, Mark can write code. He's good at what he's good at, but this is asking a lot of him."

Yeah, Geoff definitely was touchy about Mark. I couldn't resist asking. "Why? I mean code is code. Mark is a smart guy."

"Well, *smart* is a relative term."

At this point, Geoff leaned over to one side and reached into his back pocket, extracting his wallet. He dug around for a second, and then removed a carefully laminated card, which he handed to me. Turns out, Geoff was a card-carrying member of Mensa—literally, card-carrying.

OK, well that was quite a transformation—from sincere and helpful to complete horse's ass in about six minutes. I was fairly stunned, but I managed to mutter, "Well, that's impressive, but I still wonder if you're underestimating Mark."

Geoff shrugged, adding, without conviction, "Hmmm. Yeah, maybe. We'll see."

So the guy can write C++ code blindfolded, but he's got some serious limitations in the team player department. Clearly, Geoff thought Jack deferred to Mark, and it bothered him. I'd worked on enough projects to know the signs of someone who's about to go off the rails. This one was barreling along at full speed, and the bridge ahead was out.

The next morning I was sipping my coffee on my way past Mark's office and I popped in to see what he was working on. I'd fallen into the habit of doing this fairly regularly to help me get up to speed on our software. On this particular day, we were discussing the eventing model Mark had devised for our app. The early versions of Visual Basic did not allow public methods on forms, which made it cumbersome to manage MDI apps. Mark had come up with an ingenious and simple workaround, using the tag property of a hidden button to simulate raising events on child forms. It was fairly elegant, was easy to use, and did exactly what we needed—a typical Mark solution.

Mark and I were quickly building a nice working rapport. As a developer, he was more accomplished than I, but we tended to approach programming problems from a similar perspective. I agreed with his assertion earlier that our primary focus had to be on client needs, not technology. I wondered what he thought of Geoff's resistance.

"So Mark, how's the survey of the Microsoft stuff coming? Do you think we'll be able to avoid rolling our own?"

"Not without sacrificing some design requirements. Geoff is way too uptight about this. It would be cool to write our own controls, don't you think?"

Honestly, our personal enjoyment had never entered my mind as a viable factor in making the decision. Of course, as a developer, I could hardly disagree—it would be a great project. Still, I tried to put myself in Jack's shoes, saying, "Yeah, it'd be fun. But I'm not sure that's at the top of Jack's priority list right now."

"Well, he should keep it in mind. Keeping us happy is half the battle, right?" I couldn't tell if his grin was conspiratorial or sarcastic. Mark continued. "Anyway, Geoff acts like he's the only person on Earth who can write C code. Give me a break. We'll definitely get enough benefits out of writing our own to justify it. It won't take that much longer."

Danger, bridge out ahead. "Well, if it won't take that much longer, then it's an easy decision. I guess that's why Jack is trying to quantify it."

As if on cue, Jack appeared, asking, "What am I trying to quantify?" As a manager, Jack was extremely accessible. He often took the initiative to walk around the office and drop in on people. His chats were always informal, and often had nothing (directly) to do with work. Today, however, he'd walked right into the middle of a work conversation.

I answered his question first. "Oh, I was just asking Mark about the UI controls and how much longer it would take to do our own."

Jack nodded, pointed at Mark, and said, "Oh good, I wanted to talk to you about that. I don't have time right now; let's do it tomorrow morning at our 10 o'clock."

"Sure."

Then Jack turned to me and added, "Bill, why don't you stop by, too. I'd like for you to be in on this discussion."

“OK, Jack.”

“Great, see you guys tomorrow at 10.” With that, he strolled out and down the hall. I looked at Mark, but couldn’t read his expression. I offered, “Well, guess we’ll have an answer soon enough.”

“Yep, I sure hope it’s the right one.”

I hoped so, too, but I couldn’t worry about it at the moment. I had problems of my own to deal with. Part of my role at Computech was managerial, and as such I had a single developer reporting to me. Bob had been with Computech for seven or eight years. When I first came on board I remember wondering why someone with his tenure was reporting to me and not the other way around. My confusion soon cleared up.

One of our tech guys came into my office and told me that all the color toner had been drained from our new LaserJet by Bob only one day after the cartridges were replaced. The tech was understandably upset: the toner was new, it was expensive, and it was on back order. Furthermore, he implied that Bob had wasted the toner, but would not elaborate. As Bob’s manager, it fell to me to unravel the mystery of Tonergate.

I stopped by Bob’s cube. After catching up on some dev issues, I got to the real point of my visit. “So Bob, have you been printing a lot of stuff? Our LaserJet has run through a whole color toner cartridge since yesterday.”

Bob certainly had his faults, but dishonesty was not one of them. As such, he answered, “Yes.”

Clearly, loquaciousness was not one of them, either. I prodded: “Why? Why do you need to print so much stuff?”

“I was printing menus for my girlfriend’s new Thai place. Remember, I gave you one of her cards?”

I did indeed remember Bob giving me a business card when he recently described to me his new girlfriend and her new business venture. What I did not recall was any mention of Computech making material contributions to that start-up. “Uh yeah, sure, I remember that. But why are you printing the menus here?”

“It would cost us a fortune to print them at Kinko’s.”

Typically logical; Bob was nothing if not logical. He was a little weaker in other areas, like social convention or understanding implied things that are obvious to everyone else. “Right, Bob, but you can’t crank them out here at the office just because you don’t want to pay for them.”

“Why not? I thought we were allowed to use the printer for personal items.”

“What makes you say that?”

“Well, last week Mark printed out a picture of his daughter.”

"But that's just one picture. You wiped out the printer. No one in the office can use it for a week while we wait for toner."

"I didn't know I would wipe it out."

"But you did. You can't print that much stuff."

"Well, if there's a limit for personal items, you need to tell me what it is."

By now I was aggravated. "Until further notice, your limit is zero."

Now it was Bob's turn to be aggravated. "But that's not fair. Why should I have a different limit than everybody else?"

"Because everybody's different, Bob. Everything is not automatically the same for everyone."

"Well, that's bogus. I'm gonna talk to Jack about that."

I sighed. "Fine. I'll talk to him, too."

"Fine," said Bob, as he turned back to his monitor.

That had turned into a mess. Bob was by no means a superstar developer. And obviously, he could be difficult to work with. But we needed his contribution to make our dates, and as the new guy I did not feel great about alienating him. As a manager, my job was to pull the team together, not split it apart. I walked back to my office, replaying the scene in my head, wondering if I could have handled it differently. I didn't really see another way to play it, but that didn't reduce my anxiety over what Jack might have to say about how I'd handled Bob.

Luckily, dwelling on my anxiety was not an option, because I had my Wednesday 11 a.m. to get to. Most of the team would be at this weekly status meeting. Normally it was fairly uneventful, but given the tension between Mark and Geoff, and now Bob and me, I entered with some trepidation. I planned to sit next to Doug, thinking that if fireworks started going he would most likely be out of the fray. As soon as I took my seat I regretted my decision. One look at Doug and I could tell he was wrapped up tight. He licked his lips and fidgeted around, drumming his fingers on the table. His eyes darted around the room. I braced myself. If most of the people in the room, including me, were already on edge, then this meeting could quickly turn ugly.

Doug started out, still fidgeting around in his chair. "All right, I just need to clear the air because I'm having a hard time concentrating with what's going on."

Here we go. I looked around to check reactions. Everyone looked a little surprised. No telling who would be the target of this diatribe.

Doug continued. "I'm having a hard time today because last night we had a church gig and our lead singer dropped the f-bomb."

The sound of a chuckle being stifled was clearly discernable. I put a hand over my mouth, attempting to cover a smile. Doug, apparently unaware that we did not share his distress, went on to describe in some detail the career ramifications of this inconceivably careless faux pas. I looked over at Jack who, like the rest of us, was halfheartedly trying to suppress a grin. At the time, I was unaware that Jack had taken a risk by knowingly hiring a guy with long-term plans that did not include Computech. Thinking back on it, I wonder if part of Jack's grin came from realizing how big the return on that gamble turned out to be.

A couple of hours later, I was working through some code in my office when my phone rang. I picked it up. "Hello, this is Bill."

"Hey, Bill, it's Jack, you got a minute?"

"Sure Jack, what's up?"

"Can you swing by my office? Bob is here and I want to talk about this printer thing."

"Sure, I'll be right there."

I remember thinking "crap" as soon as I hung up. I should have made a point of getting to Jack and telling him my side of the story. Instead, I'd put it off and now Bob was in there lobbying. Jack and Bob had a working relationship that spanned the better part of a decade. Jack and I, on the other hand, were still in our first month.

As I crossed the threshold of Jack's doorway, he looked up from behind his desk and said, "Hey, Bill, thanks for coming." Gesturing vaguely with his hand, he added, "Have a seat." Bob already occupied one of the two seats in front of Jack's desk, so I took the other. As I sat, I nodded, greeting him with "Hey, Bob."

Not wanting to be out-matured by me, he responded tightly: "Hi, Bill."

So much for the small talk. Jack didn't waste any time, getting right to the point. "OK, Bill, so I've been chatting with Bob about this printer thing for a few minutes. He thinks you're unfairly restricting him."

I'd been thinking off and on all morning about the best way to present my case. Ultimately, I'd decided that the facts spoke for themselves. In another environment, I may have tried to be more political. At Computech, to my well-documented dismay, I was out of the loop, so it was hard to play politics. As such, I recounted what had happened earlier, pretty much without embellishment. When I was done, I added, "I also learned that for these menus, Bob used up four reams of paper. I know we've got plenty of paper around, but it just seems like, on principle, this is excessive."

Bob was ready for this one, responding, "Well, since I've been working here I always make a point of reusing my printer paper. So after I print on one side, I reload it all and print on the other side, and I know I'm the only one here who does that. I haven't calculated it exactly, but I estimate I've saved Computech at least 10 reams of paper in the past six years."

I'm not sure, but I think that my jaw visibly dropped. There was no way Jack could go for this, could he? It was just too ridiculous. But it did have an inherent logic, and Bob had been working here for a long time. I could only assume his attitude was not a surprise to Jack. Was it possible that Jack condoned it? I squeezed the armrests of my chair as Jack started to speak.

"Sorry, Bob, I'm with Bill on this one."

I let out the breath I'd unconsciously held in and relaxed my grip. Bob started to object, but Jack cut him off. "No one else has this problem. Everyone but you seems to understand the implied limits. You asked for a limit, and you got one, so live with it."

Bob wasn't going without a protest. "That's bogus. It's not fair."

Jack responded with a phrase I would hear him use many times over the years: "If life was fair, there wouldn't be rich people." It was kind of an odd phrase, considering most of our clients were Wall Street financiers, but it got his point across and effectively ended the protest.

That was pretty much the end of our meeting. Later that day, I popped in to thank Jack for supporting me. He responded, smiling warmly: "Hey, don't mention it. I know Bob can be a handful. You did what you had to do."

I asked, "So you give the handful to the new guy to get him out of everybody else's hair?"

Jack grinned. "Or maybe I give him to the new guy to see how the new guy handles him."

Now it was my turn to grin. "So how'd I do?"

"You passed."

Over the years, I've often wondered about Bob and why Jack kept him on board. Sure, he was a handy litmus test for new managers, but that hardly justified his salary. So what was Bob's saving grace? What was the one trait that made him a crucial cog in our well-oiled machine? Was it his mediocre code? His abrasive personality, perhaps? All of the above? I think the answer is none of the above. Bob was not a crucial cog. The biggest thing Bob had going for him was that he was part of our team. Bob consumed a big chunk of Jack's time, and Jack never blinked. Jack was there to support us, to facilitate our work. That was the message he sent to Bob, and that was the message he sent to our team.

I talked earlier about the loyalty Computech engendered. It doesn't come from executive emails or dinners on the company dime. Loyalty is a two-way street. If a manager is committed to working with his most difficult employee, what message does that send to everyone else on his team? I remember years later, after Computech had been bought, and then its new corporate parent had itself been bought by a competitor, we went through a round of layoffs. The layoffs ended in December, and by the following February we were hearing from our new management about building employee pride and boosting morale. Pretty typical stuff in the world of corporate mergers and acquisitions.

I thought about Bob, who was released in the pursuit of efficiency. In a way, I was glad to be rid of him. Man, he was difficult. It was hard to argue the business case for letting him go. Or was it? Those daily morale-boosting emails we received were widely derided, considered transparent ploys to keep us docile and productive. Resumes were drafted, production diminished. Morale was low. And I thought about what would happen if Jack sent a similar email, asking us to buckle down for the good of Computech. I think we would have believed him, and done what he asked. We would have sacrificed for him, because we knew he would do it for us if we needed it. And that's what Bob brought to our team: proof that we really were committed to teamwork. He was, perhaps, Jack's most visible vehicle for showing that we were all in this thing together.

The next morning, I dropped into Mark's office a little later than usual. After discussing a few technical issues with the new build, we made our way down to see Jack. As Jack welcomed us into his office, we settled into our usual seats and began chatting about the Eagles. Jack was a huge Randall Cunningham fan; Mark and I thought his best years were behind him. That conversation (which he had several times a week) ran out of inertia and Jack switched to the topic at hand.

"All righty, well, let's talk about this UI stuff. Mark, what do you think? How much can the Microsoft stuff do for us?"

Mark responded: "Yeah, I've been looking at it for a few days and it looks like it's gonna be somewhere between one-half and two-thirds of the requirements we've been asked to do." He went into a brief description of the functionality that we would not be able to implement. Jack responded with some specific questions about which design requirements would and would not be preserved. Mark had a good handle on the scope of the work, and addressed each question. After a few minutes, it seemed quite clear to me that the Microsoft controls would handle at least two-thirds of our requirements, and probably more like 75% or 80%.

If Jack noticed this discrepancy, he didn't acknowledge it. Instead, he said to Mark, "OK, what I'd like to do is take one control and write it from scratch as a pilot. We can see how it goes, and then make a decision on the rest of them." The logic of this as a managerial decision was clear. In fact, I had the distinct impression that Jack had made that decision before we ever walked into his office that morning.

I don't know if Mark had the same impression, but I do know he was pleased with the decision. "Cool. Which one should we do?"

That was an easy one. Makes sense to do the simplest one first, eliminate complications from your pilot. My guess would be the button control. Jack, however, screwed up his face as though this detail had never occurred to him. He asked, "I don't know, which one do you want to do?"

Mark beamed. "I think the virtual listbox would be a good one to start with. It's complex, but we'll be able to use it everywhere, so we get the most bang for the buck." He paused for a second, and then added, "Plus, it'll be a fun one to write."

Jack couldn’t resist a grin at Mark’s enthusiasm. “OK. Virtual listbox it is. Can you give me a dev estimate by the end of day tomorrow? Also, I really need you to work with Geoff on this one.” Mark’s expression turned dour. Jack continued, “Don’t worry, I’ll talk to him. But we need to take advantage of what he can do if we’re gonna be able to deliver these in a reasonable time. Otherwise, we’ll be stuck with the Microsoft stuff.”

The three of us chatted for a few more minutes. After we’d left Jack’s office, I turned to Mark and said, “That’s great, man, you get to do the V-List.”

Mark was back to beaming. His face turned serious and he said, “Yeah, and I’m really gonna nail it so he lets me do the rest of them.”

Of course, Mark was only half of the equation. Geoff’s cooperation was critical. Jack really needed to work some more magic. Later that day, Jack caught up with Geoff for a face-to-face chat. When an ego needed soothing, Jack preferred the one-on-one format. As such, I wasn’t in Jack’s office for this particular session, but having talked to Jack about it later and also seeing firsthand how he operates, I can pretty fairly reconstruct it. It started with Jack playing into Geoff’s ego, making him feel like his opinion mattered, saying something like, “Geoff, thanks for stopping by. I just wanted to pick your brain a little more about these UI controls. What’s your feeling on the work required to get them done?”

Geoff, still on the defensive, replied, “Well, like I said, I think it’s gonna be a lot of extra work. I’d estimate at least three months.” He paused to maximize the effect of this shocking news. Jack, unflappable as always, nodded acceptingly, so Geoff continued: “And I don’t really see the point. We don’t get that much benefit for the extra dev time.”

Jack tried to diffuse Geoff’s stated objection. “Yeah, I hear you. Let’s not worry about the dev time for now. I’m more interested in the feasibility.” Then he went after Geoff’s primary, if unstated, objection. “The thing is, I think this would be a good learning experience for the team, particularly for Mark. We both know he’s a good programmer, but you are my top C++ guy. And I think it would make the whole team stronger if we could take your skills and spread them around to the other guys.”

Geoff, somewhat disarmed, replied, “Well, sure, it’s always good to have critical knowledge shared among the team, but I’m not sure this is the time to do it.”

Here, Jack could call on his own experience. “Well, the thing is, there’s never a good time to do it. I’ve never been on a project where there was extra time to devote to training. However, I’ve learned that squeezing the training in there almost always pays for itself before the project is over.”

Geoff, feeling secure about his own skill set, was willing to cede to Jack’s expertise on project management. “OK. If you think the delay is OK, then that’s your call.”

With Geoff more open to the change of plans, Jack went after his complete buy-in. “Yes, it’s my call. You don’t need to sweat that. What I do need you to sweat is helping out Mark when he needs it. Like I said, we need to spread your knowledge. What do you think?”

“I mean, I’m OK with it, Jack. I’m just not sure how effective it will be.”

“Well, how about this? I’ll set up an hour a week for you two to get together around this stuff, at least to get the ball rolling. Let’s try that for a couple of weeks and see what happens. And try to be patient with Mark; he’s used to knowing more than the other guys, so it can be hard for him talking C++ with you.”

Jack had hit the right buttons. Geoff beamed and nodded OK.

“Thanks, Geoff, I appreciate it.”

“Sure, no problem.”

A few days later, the first of these meetings took place. Mark and Geoff started out with a weekly meeting to write a listbox in C++. At the peak of the project frenzy, they were having three or four impromptu meetings per week as they wrapped up the listbox and moved on to buttons, text boxes, and a few others, totaling half a dozen custom user controls by the time this project wrapped up. Geoff’s prediction of a three-month delay turned out to be overstated by two months, mostly because the experience writing them made each successive one go faster and the knowledge that was passed from Geoff to Mark continued its dissemination to the rest of the team.

As for me, the budding VB guru, in retrospect I can see that my worries about not being plugged in were misplaced. I actually had a pretty good handle on the dynamics of our team. Yes, that’s right, there was no loop. I had inklings of this early on, but it took me the better part of a year to fully accept it. Sure, it sounds nice. A team focused on their work; supporting each other in pursuit of a single goal. Yes, it really happened. Not necessarily the 10 best programmers who ever worked together. Instead, each of us brought something to the table. We each had something to contribute, guided deftly by a leader sensitive to people, aware of their strengths, and accepting of their flaws.

So you’ve probably guessed by now that the Computech rewrite from QuickBasic to Visual Basic 3 was a resounding success. Indeed, it was. We had our bumps along the way. Turns out Doug did hit it big. Well, biggish. His band went on a U.S. tour. Luckily, it was toward the end of the project, so when he left we already had people lined up to fill in for him. Bob caused his share of friction, and some of his features did not make it into the first Windows release. But these were minor flaws in an otherwise overwhelming success.

Ten months after that day in the conference room, Computech’s early adopters began receiving a Windows version of our software. Feedback was almost unanimously positive, reports of problems were minimal, sales skyrocketed, and executives were happy. The development team received its share of accolades. Even to this day I’ll occasionally reminisce with an old Computech veteran, and they’ll point out how lucky we were to get so many fantastic developers at the same time for a single project. I always respond that it wasn’t luck and it wasn’t fantastic developers; it was a beautiful team.