CHAPTER TWENTY-FIVE

Welcome to the Process

Step Inside, Step Inside, and See the Show

Ned Robinson

IT WAS APRIL 2001, AND THE GRAY SKIES OF WINTER IN NEW YORK WERE GIVING WAY TO THE CRISP blue skies of spring. A 20-year programming veteran, I had spent the previous five years running the software division of RePlay Technologies, a music software company. The company had done relatively well for a small start-up, but after five years of almost breaking even, RePlay was sold and I was out of a job. So here I was, back in the financial district of NYC looking for a job.

My first few interviews proved that the American corporations had not changed their software development practices over the previous five years. They were still paying little attention to formal processes and good project management. Every company I interviewed with sang the same refrain: work 60 to 70 hours a week, projects will succeed because of your superhuman efforts (red cape and beeper will be handed out on first day of job), and yes, expect the sponsor to be changing requirements constantly. When can you start?

The problem was that I no longer wanted to be a superman developer. I had done that for too long already and was burned out. I hadn't minded when I was still in my 20s and trying to make a name for myself in the industry. Whatever the project, whatever the obstacles, I had spent many sleepless nights writing line after line of code to accommodate incomplete and half-articulated and thought-out requirements that lasted only until the boss came in the next morning with some new ideas that wiped out 80% of the work I had just done. The adrenaline rush of handing my boss a working program with the changes he had requested the night before as he left the office had long ago lost its luster. I no longer wanted to live in the world of constant scope creep. I no longer wanted to be part of a team of half-asleep, over-jolted programmers.

All of these thoughts were in my mind as I entered the lobby of the World Trade Center, Building 2, for an interview with Benchmark, a small software company that specialized in desktop stock evaluation software for asset and portfolio managers. I rode the elevator up to the sky lobby on the 78th floor and entered the office.

The offices were beautiful. Spreading out on a portion of the 78th floor and the entire 77th floor, everywhere I looked I saw unobstructed views of the most incredible New York City vistas. As I walked up to the reception desk, I could see the Statue of Liberty from the window on the right. This was a pretty cool environment, I thought, from a physical standpoint. I was hoping that the job lived up to the surroundings. After checking in with the receptionist, I awaited the arrival of the software development manager.

When Curt arrived at the reception desk, I was quite surprised (and equally excited) by his dress and demeanor. He was dressed in black jeans, a button-down collared black shirt, and black army boots. He had his straight black hair pulled back in a ponytail and there was a 2-inch streak of purple hair on the right side. An earring completed the picture. All of a sudden I had hope that this job might offer something different from what I had been imagining.

Curt and I went down to his office on the 77th floor. His office was littered with all kinds of hardware, hard drives, motherboards, memory chips, and so on. Our conversation started off easily as I related my work history. It turned out that Curt was also a musician, having played cello in school and now playing bass in a punk band. We talked about music and about how I developed a suite of music processing algorithms for RePlay Technologies. We shared war stories about writing code in beta versions of Microsoft compilers, about writing 16-bit code in a limited memory model, and how to handle users who always seemed to wait until you were just about finished to ask you to add just a few more fields (to all the screens and reports). We even spent a few minutes discussing the pros and cons of FORTH and reverse polish notation.

He then told me about the development team that he was managing and the process reengineering that he had already started. This is getting interesting, I thought. He told me about getting his undergraduate degree at Carnegie Mellon, the home of CMMI (the Capability Maturity Model Initiative) and his efforts to date to bring the company to Level 2 compliance on the CMMI scale. After he concluded his very brief introduction of CMMI, he said the following sentence that won me over even before the job was offered: “If you ever have to work more than 40 hours a week, then I have failed at my job as manager. You have not failed at yours. My job is to hire good people and stand back and let them do the job that they were hired to do. For software developers, this means department or companywide adoption of time-proven processes and methodologies that clearly define all project work and scope and have a strong change management component."

Wow! Was the ground going to break open and swallow him up for blasphemy? Did he just say there is a way to manage scope creep? Could any more noble words come from a boss's mouth? Did he really have senior management buy-in for this? While I still had a lot of questions, so far this looked like the job for me. Over the next couple of days, I met the members of the development team, and took a technical test, a personality assessment, and a drug test. Within a week, I had a new job.

I was the fourth developer on the software development team. The team worked well together and each member had a particular area of strength that the other team members relied on. Mary was great at developing user interfaces. Dmitri was the bit guru and wasn't scared of getting down and dirty with the code. Peter liked decomposing the business requirements and creating the software development plan. I was intimately familiar with Windows internals, object-oriented design, and multitiered architecture. We all had our strengths and would prove to be the nucleus of a growing team that worked well together and loved writing code.

The team also worked well with the other departments in the SDLC: business requirements, quality assurance, project management, operations, and frontline business users. They worked so well together that one of the developers married one of the QA testers. In the spirit of continued family harmony, we did our best not to put them on the same project!

There was a spirit underlying the entire company that created a really positive work environment. Many of the approximately 200 employees had been with the company for five to 10 years. was All of them were committed to seeing the company grow and to feel like they were a part of it. This was promoted from the top level of the company down. Customer milestones were tracked, and whenever a new milestone was hit (every 100 new desktops or 10 new customers), each employee received a wooden plaque and a crisp $100 bill. For those who had been with the company for a long time, their cubes were proudly tiled with plaques from floor to ceiling.

Things were good at Benchmark. While the company had been bought by a large financial information firm, Benchmark was still operating pretty much independently, although rumors abounded about being folded into the larger corporate family. But, at the moment, Benchmark would be able to follow its own processes and methodologies.

My first project was to write a program that continuously monitored critical backend data collection processes. The program was to check that data was flowing into the company's servers from a number of data providers and that the data was being automatically converted, saved, and made available for redistribution to our customers. It also had to make sure that the delayed security price request system was working and providing accurate quotes. This Watchdog program updated a frontend monitoring program that Dmitri had written and that ran on an IT workstation 24/7.

As I started to write my use cases, I began to understand why Benchmark had such a bad record of delivering quality software on time and to spec. In order to write this Watchdog program, I had to know what each system being monitored was supposed to do. As I looked for information on each system, I discovered that many of the systems had no documentation or that the documentation had been a preliminary specification that in no way matched the final deliverable. For example, the datafiles from one vendor had to be converted to a specific layout with a checksum added to each row and to the end of the file. Unfortunately, the final data layout of the imported file had been changed by the vendor and fields were added to both the middle and end of the data structure. Since I had to check specific fields and compute a checksum, I had to make sure that I was reading the right fields. To make matters worse, the developer who had originally written the import program was no longer with the company. I tried talking with Nancy, the head of the department that managed the data feeds. Unfortunately, she had no patience for requirements gathering and told me that she didn't have time to discuss it and to just figure it out myself. In the end, I had to do just that: open the import source code and decipher what was going on.

I continued to hit these types of walls as I worked to complete my use case analysis. The only smooth part of the requirements gathering was when I had to figure out how to interface to Dmitri's monitoring frontend. When I asked Dmitri how to do it, he sent me a link to the complete and up-to-date API spec! With a complete requirements document in hand, I was able to finish the Watchdog program a few days before my required delivery date.

As this project was coming to a close, Curt assigned me to work on another internal project for Benchmark. One feature of the software was to track the comings and goings of all the top Wall Street executives: what companies they worked for, their previous and current titles, promotions, salaries, bonuses, and so forth. Curt gave me a rough draft of a requirements document that Nancy's team had given him: a one-page brief description and a couple of hand-drawn screens. A kickoff meeting was scheduled for the following week.

As the meeting began, all of the participants were introduced. Nancy had brought six members of her staff to the meeting while all the other departments had one or two representatives. One of Nancy's assistants introduced the purpose and scope of the project. After they finished, Curt asked Nancy when her team was going to deliver the completed requirements document. Nancy said that she felt what she had delivered was enough to understand all of the requirements of the project and that they were too busy to do any more writing or analysis. Since there were no business analysts available to work on this project, the software developer would just have to figure it out himself. Curt and I looked at each other with shock and dismay. We couldn't believe that she was serious. Curt made some comments about what Nancy had promised to deliver and Nancy made some counter comments about how the software developers had to be nimble and be able to go with the flow.

Curt stood his ground and told Nancy that the project would be on hold until more requirements were available. Curt wasn't going to cave in like other managers I have had in the past. At that point, the project manager ended the meeting and said that it would have to be continued after the issues with the requirements had been discussed offline. As we left the conference room, I smiled as I realized the street credit that Curt had just gotten.

As summer 2001 was coming to a close, I felt at home at Benchmark, like I had always been a part of the team even though I had worked there for less than six months. As I arrived at work one morning in late August, I walked off the elevator and was greeted with an incredible view of the Statue of Liberty framed against a crisp Carolina Blue sky. I had my headphones on and was listening to Gershwin's "Rhapsody in Blue." That was one of those moments that I will never forget: the music, the view in front of me, the blue skies above. I sighed and walked just a little lighter to my desk.

Less than two weeks later, that view would be gone forever; even the possibility of stumbling across the nexus of that view and that music, at the same time in the early morning light—that possibility would be gone and the only artifact remaining through eternity of that experience would be the recording of it that is burned in my memory.

It was 8:30 on the morning of September 11, 2001. Some of the team was already in the office and working on the day's tasks. I was going over a list of questions I had on the Nancy project. It was a quiet morning on what I thought was going to be a quiet day. I took a short break and checked in on Barry Bonds and his quest to break Babe Ruth's home run record. I was just about to get my second cup of coffee for the day when all of a sudden we all heard a loud explosion. Peter and I walked over to the window, and as we looked out, a large plume of fire shot past us, followed by smoke and ash. Due to our viewing angle (we were looking out east, over the Hudson River) we couldn't see what had caused the explosion or the fire shooting past us. We waited a couple of minutes, waiting for an announcement from the building staff, but nothing came. Alice (a BA), Peter, Megan, and I all sat near each other, and after a few minutes of discussion, we decided to start walking down the stairs in the fire escape, figuring that we would get some direction soon. As we entered the fire escape, it seemed that a lot of other people had had the same idea. The speed of our descent on the stairs slowed, but did not stop as everyone walked down in an orderly fashion. Some people had handfuls of tapes, some had tapes and the tape backup units as well. Many others were carrying piles of CDs and other personal belongings. I was carrying my laptop in my shoulder bag.

As more and more people entered the fire escape, the going got slower and slower. The apprehension around us was made all the more eerie by the fact that we still did not know what had happened. Was it a terrorist bomb? Was it some kind of Con Ed situation like a manhole cover exploding? Whatever it turned out to be, not one of the thousands of people on that staircase would have thought that it was caused by an airplane flying into

Tower 1, the next tower over. Nor did we suspect that a second plane was heading for Tower 2 and was about to crash into us.

The four of us stayed together, helping to keep all of our spirits high. After about 45 minutes, we had gotten to the 50th floor. We faintly heard an announcement being made in the hallway, so everyone emptied the fire exit to hear. The fact that there was no PA system in the fire exit must have been a design flaw, no?

The crackling male voice told us to remain calm, but no information was given about the cause of the explosion. The voice just reassured us to remain calm, that the situation was in Tower 1 and was under control, and, here's the kicker, that we should all head back to our offices and wait for further instructions.

Here was a real dilemma. We were all trying to stay organized and follow the instruction we were given. Unfortunately, at a gut level, this instruction didn't make sense to anybody. So, what to do? Do what our gut was telling us to do and get the hell out of there, or do what we were being told and go back to our offices? Megan and Peter stepped off by themselves to discuss it, which left Alice and me to come up with our own decision.

After going back and forth on what the right course of action should be, we finally agreed that we should go downstairs, get a cup of coffee, and see what was going on. Alice and I headed back into the fire exit, which was now empty as everyone else was still deciding what to do. We were actually able to run down the stairs. We ran and ran, and then suddenly felt a huge explosion. The stairs started to sway back and forth, a good three feet in either direction. I grabbed on to the railing, half expecting it to pull out of the wall as the stairs were turned into a pendulum. For some reason, I had a vision of Shelley Winters in the *Poseidon Adventure*, holding on to a steel ladder that just pulled out of the wall when she grabbed it before being dumped into the water. But, to my great relief, the stairs in Tower 2 held and the swaying eventually stopped. I noticed that we were on the 43rd floor. To my amazement, there was absolutely no damage to the stairs. The lights were on, the staircase and railings were solid, and there was not a chip of paint or plaster anywhere to be seen. Alice spoke first and said that we needed to keep going. I agreed and we continued to run down the stairs until, once again, other people had entered the fire exit and the descent again slowed to a crawl.

It took us about forty-five more minutes to get out of the building. As we entered the World Trade Center shopping pavilion, fire department and other emergency service personnel were lined up every 5 to 10 feet, creating a corridor for us to walk through up and out by Building 5 (or was it 7?). They were proud to be doing their job, helping us get out. Every day I have prayed that they, too, were able to get out before the building collapsed.

All in all, the company of 200 employees lost four people that day. We had all felt like one big family. The beautiful office space that the family had struggled to attain, which was a symbol of our success and unity, was gone. We were homeless, made that way by a tragedy that will never be forgotten. And we had lost four comrades. But, like any good team, we pulled together and worked tirelessly to get the company back on its feet, to support each other emotionally and spiritually and in any way that we could.

There are many stories of escape from Tower 2 and I would like to share a few team members' efforts to survive. I also want to mention that when the plane hit Tower 2, it had crashed right into our conference room on the 78th floor. This conference room was the showpiece of the office. The side facing outward was all glass and there was an incredible view of the southern tip of Manhattan. It was such a special place that our parent company would frequently use the conference room when making important presentations. That morning, two people from the parent company were there early to set up for a presentation later that day. Unfortunately, they did not survive the attack.

Aparna was a member of the QA team. She was eight months pregnant on 9/11. She had decided to stay near her desk when we heard the first explosion. When the plane hit Tower 2, the ceiling panels on the 77th floor fell, filling the entire space with dust and debris. Aparna could not hide under her desk to escape because of her pregnancy. All she could do was stand there and get showered with debris. When the dust had started to settle, she began to call out for help. Rick, the manager of the QA department, and Tony, the manager of data operations, both came running. They helped Aparna to the fire exit and the three of them walked down the 78 flights of stairs together.

When they got outside, EMS immediately took Aparna to a local hospital to ensure that she and her baby were OK. They were, and one month later, Aparna had a healthy baby boy. Rick was covered in dust and EMS gave him oxygen. He was sitting on a curb on Church St. with the air mask on. All of a sudden, the police started yelling for everybody to run as fast as they could. The building had started to collapse. Ripping the oxygen mask off, Rick started running for his life. He ran down John St. but there was no place to enter as most businesses had closed by then. Suddenly, he saw an open bar and grabbed the door handle and pulled himself inside. As he closed the door, the rolling cloud of the remnants of Tower 2 blew past, leaving Rick safe inside the door.

Bernie was the director of IT. He had come in that morning and gone directly to the conference room on the 78th floor to help the visitors prepare for their presentation, making sure the advanced A/V system was working. With everything in working order, he accidentally left his briefcase on the 78th floor and returned to his office on the 77th floor where he was when the first plane hit Tower 1. He spent some time coordinating with people on his side of the building and was getting ready to leave when he realized that he didn't have his briefcase with him. He picked up the phone and called Violet, who sat next to the conference room.

"Violet, this is Bernie. I'm looking for my briefcase. Can you tell me if it is in the conference room? And, by the way, what are you still doing there?"

"Bernie, yes, your briefcase is in there. I can see it from my desk. I am just going to put the phone system on night standby and then I am going to get out of here," Violet replied.

"OK, thanks," Bernie replied. "I have to call my mother to let her know I am OK, and then I will get the briefcase and head on out myself. Thanks."

Bernie quickly spoke with his mother, and as he said goodbye to her and hung up the phone, Violet, who was still at her desk, saw a dark shadow cover the conference room window and then "POW," the second hijacked plane crashed through the window and Violet, now frozen in her chair, watched in shock and horror as the plane's wing flew right past her, just a few feet from her face. Luckily for Violet, she was able to get herself moving and headed right for the exit stairs.

Bernie also headed to an emergency exit. The first one he went to was jammed shut. He tried to push it open, but something was blocking the door on the other side. He ran down the hall and found another exit. He pushed, and this time the door opened easily. However, the stairwell entrance was flooded and some kind of liquid was leaking down from the ceiling above Bernie's head. He thought it smelled like kerosene and assumed that a pipe in the heating system had cracked. He didn't know that he was being dowsed in jet fuel.

Another worker, Anthony, came running into the stairwell, holding a flashlight in front of him. Together, he and Bernie ran down the stairs. As Bernie grabbed the railing, he noticed it was wet and sticky. He assumed that the kerosene was leaking on the railing as well as on him. They ran down about thirty floors when, all of a sudden, the lights went out. Total darkness. Anthony switched on his flashlight and the pair had to slow their pace for another 10 floors until they came to a place where the stair lights were working again. In the new light, Bernie took a moment to look down at his hands. They were covered in blood. The liquid that was on the railing on the upper floors had not been the same kerosene that had soaked through Bernie's shirt. It had been human blood.

Bernie and Anthony had made it to the lobby of the WTC and were now running out to the street. An EMS worker came right up to Bernie who was soaked in blood and jet fuel. Immediately, the EMS worker made Bernie take off his shirt. As he was lifting the shirt above his head, they heard the police yelling "Run!" Bernie started running, but he didn't have as big of a head start as Rick did. Instead of being able to find shelter, Bernie just kept running, even as the debris cloud caught up with him and coated him the same way it coated Lower Manhattan.

As that September day went on, I was able to speak with all the members of the software development team and confirm that everyone was safe, alive, and accounted for. I was also asking for volunteers to go to our small Philadelphia satellite office to rebuild all the backend data servers that had been in New York and to establish the numerous operational procedures that were involved in the nightly push of data to our clients.

Unfortunately, at 12:30 that night, just as I was getting ready for bed, my phone rang.

"Hello, Ned?" the caller said. "I am a friend of Barbara's husband. You work with Barbara at Benchmark, right?"

"Yes, I do," I replied.

"Well, no one has seen or heard from her since the attacks this morning. Her husband wants to know if you might have seen her since then."

"No," I whispered helplessly. "I saw her at the coffee machine very early in the morning. But our desks were on the opposite sides of the building, so I didn't see her after that."

"OK, thanks," the friend said. "Her husband just wanted to call everyone possible."

I hung up the phone. Barbara was never found and was eventually counted as one of the four Benchmark employees who were killed that day.

By the next day, 15 people from Benchmark had arrived at the Philly office. This office was home to a small group of software developers who were responsible for writing and maintaining the frontend application. There was one conference room which was going to be repurposed as a data center, and many developers were already sharing small cubicles. The office quickly began to look like a small college campus during finals week.

The goal was that by the time the stock market reopened the following Monday, our software would be fully operational and we would be able to push data down to all of our clients. That meant that there was less than a week to get data feeds and servers installed and configured and to establish the three-man nightly operations center that was required to support the software.

Curt and Dmitri represented the software group. They worked tirelessly over the next week making sure that all of the custom software was loaded on the servers and correctly configured. The rest of the software team that had stayed in New York were communicating with them constantly to help stress-test the data feeds as they became operational and to ensure that the customers would be able to use the software. Sleeping was rare, and so were hotel rooms. It felt like we were working at NASA during a lunar mission! Pizza boxes were stacked floor to ceiling in any corner that wasn't occupied by a data server or a developer with a laptop.

By the time the stock market opened the following Monday, our software was ready to go, albeit with a few Band-Aids in place. However, our clients were happy that at least one of the data providers was working.

Dmitri stayed in Philly for a few extra days to do an inventory of our backup source code servers. We had to make sure that all production software could be compiled and run according to specification. Dmitri called me at home with utter despair in his voice. He told me that while the SourceSafe server that contained all of the company's documentation had been backed up nightly, the server that contained all the source code had never been added to the nightly backups. Oops! No one had bothered to test the integrity of the server backups. All of the software that Dmitri had written at Benchmark for the previous seven years was gone.

We spent a lot of time over the next few weeks trying to figure out what source code we did and didn't have. We checked our home computers and any laptops that we had with us. Fortunately for Dmitri, his last big project had been saved by one of our process improvement initiatives. We had instituted peer reviews between the New York and Philly offices and had recently sent the Philly group a large project that Dmitri had worked on. The source code was safe on the Philly development manager's computer.

In the end, senior management decided that since so much source code was lost, only production software that was in immediate need of an upgrade would be reproduced. This allowed most of the software team to begin working on new development work. The only question now was where we would be working from.

A few weeks after 9/11, the entire company got together at the Union Club in New York City for a company meeting. This was the first time we had all gotten together after the attacks. In a way, the Union Club's requirement of formal dress helped to create an atmosphere of importance. Dressed in our Sunday Best, we came together and together mourned our lost coworkers. One of the people who had been killed was George, a salesman with the company for many years. I was speaking with one of his fellow salesmen who said he had been at the dentist that morning and, as he was sitting in the dentist chair, was in utter despair as the TV reports started coming in. All he wished for was to be back at the World Trade Center with his team. He had spent the previous few weeks feeling guilty about not being there. His thinking was that if he had been there, perhaps he could have saved his friend. I have heard the same sentiment from many people: people who were just getting off the subway, people who were still in their cars driving to work but could see the planes as they attacked us, people who lived just down the street in Battery Park. It seems irrational, but people want to feel connected in whatever way they can.

At the Union Club, Nancy spoke with Curt and me about permanently sending the developers to work in the Philly office until a new location could be found in New York. She didn't believe that developers could be productive working at home and that they needed to be closely supervised. Curt told her that the conditions in Philly were extremely overcrowded and that he wasn't going to have the developers share hotel rooms for an indefinite period of time. He wouldn't give in, and Nancy walked away frustrated and angry.

It turned out to be six months before we were all relocated together in our new office space. During this time, everyone worked from home but remained highly productive, and we constantly communicated with the rest of the company. We IM'd each other, teleconferenced, had impromptu meetings in New York City restaurants like America and Zen Pallate in Union Square, and did whatever we could to stay productive. I even had a three-hour meeting in Starbucks with one of Nancy's assistants to try to hash out the requirements for the stalled project. It just so happened that the assistant had some downtime. Nancy had told her to work on the requirements if it wouldn't interfere with other activities. By the time we moved into the new space, the requirements were complete, inspected, and approved and the coding was 50% complete.

At the end of February 2002, we settled into our new office space. We had the entire 5th floor of what used to be a dot-com start-up company in the West Village. There was a foosball room, a yoga room, and lots of curved walls and oddly angled corners. It turned out that there had been a bar in Philly that had a foosball table and some of the people who had worked there got quite good at it. The yoga room was turned into a meeting room but the foosball table stayed!

Another change was in store for us. Curt had made the decision to return to Carnegie Mellon to get more involved with CMMI and process engineering. He resigned and I was given his job. We also expanded the team to include two new developers and one consultant. The team was changing right before our eyes. But our shared experiences of the previous six months had created an invisible bond between us that could never be changed. The way we thought about each other, the way we worked together, the way we could look at a problem and know the solution the other person was thinking of without saying it—no, nobody ever talked about it, but we all felt it and honored it.

One department benefit that I instituted right away was for each developer to be able to work from home one day a week. We had proved that we could be just as productive from home as in the office. The only constraint was that we had to do it on the q.t. because one of the other managing directors did not agree with this policy (guess who?).

The team continued to work hard. Now that we had some unplanned time, we took the opportunity to create a strong code foundation. Each developer was responsible for writing reusable code and, for every project he worked on, he had to create at least one reusable component. This could be a backend processing DLL or a user interface widget. At our weekly department meetings, one developer would have to present his contribution to the rest of the group. He had to include code samples on how to use his contribution as well as showing the group the code he wrote to create his object. As a senior developer on the team, it was soon Dmitri’s turn to make his presentation.

Dmitri resisted. He came to see me and told me that he wasn’t comfortable speaking in front of the group. He had a thick Russian accent and felt that he couldn’t communicate that well and that he would ruin the meeting for everyone. I said, “Dmitri, you are always meeting with one or two people, showing them how to write code or helping them debug a problem. Everyone understands you just fine. Don’t worry, these are your friends!” We spoke some more and I told him that if he really didn’t want to do the presentation, he could skip this round. He thought about it and eventually agreed to give it a try.

The day of the presentation came. Dmitri got so tied up with making his handouts that he was late to the meeting. We sent out a search party and found him at the copy machine, battling the automatic collator. The search party helped him clear the paper jam and finish making the copies. They all came back together and Dmitri began his presentation. He had written a DLL that standardized the format of error logs, and the output could be directed to either a text file, the Windows Event Logger, or both. This was something that the developers really needed and they were anxious to start using Dmitri’s DLL. As he began writing his sample code to his interface with the DLL on the whiteboard, you could tell he was nervous. He spoke with his back to the group and his handwriting was hard to read. He stumbled through his presentation and, thinking he was finished, sat down, visibly perspiring, and asked whether there were any questions. I am not sure who started it, but there was a look of utter despair on Dmitri's face and we all just started to laugh. Even Dmitri started to laugh. And that was enough to get rid of all the tension in the room. And the meeting changed from a formal presentation to just another session where Dmitri was showing the rest of us how to do something that he knew backward and forward. The meeting continued, and before we knew it, we had run out of time and the next group was knocking on the conference room door to get us out.

Megan had the same aversion to speaking up in large groups that Dmitri had. She was one of the QA testers. She had the dogged determination needed to uncover even the most obscure bugs. When she walked over to the development area, the programmers took cover. Unfortunately, she was not much of a contributor to inspection meetings. One day I happened to be sitting next to her at a requirements inspection meeting. We all had our copies of the document. When I glanced down at Megan's copy, I was impressed that she had red marks on just about every paragraph! She had really taken the time to read this, I thought. However, she didn't say anything during the meeting. I figured that she had her reasons and didn't say anything at the time.

Later that day, I walked over to her desk.

"Megan, you have a sec?" I asked.

"Sure. What's up?"

"I was just curious as to why you didn't bring up any of your comments at the meeting this morning."

I could tell she was getting very uncomfortable with the conversation and didn't want to answer me. Since I wasn't her manager, I decided to not push the issue with her. But I did speak with her manager later on and the two of us began to gently encourage her to be more assertive.

A few months later I was at another requirements meeting with Megan. We were quickly moving through the document when all of a sudden Megan raised her hand and said, "Excuse me, I think I found some mistakes on the previous page. Does anyone want me to point them out? I don't know how important it is. I'm sorry for stopping everyone." We all turned back to the previous page and Megan showed us the two mistakes she had found. They were two small, non-critical errors, but everyone was glad to have them pointed out. "Great catch, Megan," I said. She started to apologize for finding them and wasting everyone's time, but I quickly stopped her: "No, you were 100% correct to point them out. Remember, we are so much better off finding all the defects now rather than waiting until development or testing. Good job." Over time, Megan became respected as the most thorough document inspector in the QA department.

As the group manager, it was important for me to recognize that everyone had different strengths and desires when it came to writing code. Mary was excellent at creating user interfaces. Peter was our process person. Together, Peter and I went through all the nooks and crannies of the Rational Unified Process and created what we called the Benchmark Unified Process. This project lasted a few months and the end result was a complete road map that any developer could follow to ensure that his project followed the software development life cycle process and had all the needed documents and deliverables. This became an invaluable part of the department's assets and the core for writing quality software. Our adoption and adherence to this process and the focusing on each other's strengths, not weaknesses, was the main reason why, over the next three years, the software development team had a perfect track record of delivering software on time and on budget and with the features promised. We were the department that everyone wanted to work in and be a part of. I had to keep a list of people who wanted to transfer into my department.

One day, my boss, Bernie, came to see me. "Ned," he said, "I have a favor to ask you."

"Sure," I replied. "Go ahead."

"There is a guy over in the IT department that has been writing little utilities for us in VB script. He's pretty good at it, too. I just did his annual review and he is bored working in IT. I think that if I don't involve him in more development work, he might leave the company."

"Bernie," I said, "do you realize that I already have official requests from five other people to transfer into the development group? Are you going to open up six positions for me?"

"Well, of course not," said Bernie. "But I don't want to lose Doug. Give him a regular interview and if you think he is a good match with the group, I'll work out the transfer myself."

"How about if I do take Doug I get to also take one other person from the top of the list. You know these are all junior programmers and it takes time to train them. We can have them sit with a more senior developer for a few months and they'll get some good experience. Plus, if it seems that they are not going to be good developers, we will find it out soon enough before any real harm is done."

Bernie said: "I can't make any promises now. See how it goes with Doug. If it works out, I'll see what I can do to open another job req. OK?"

"Fair enough," I said. "I'll meet with him and get back to you."

By the end of the month, the development group had two new junior programmers, Doug and Henry.

Although we were successfully adopting and adhering to our process and proving beyond a shadow of a doubt that software process does work and that the more time spent planning upfront guarantees excellent results, there were still one or two senior managers who continued to fight the process whenever they had a chance. Mary had been working on a group of reports for Nancy that “somehow” continued to fly under the process radar. One of the outputs on the report was a custom calculation. We had no specs on how the calculation was supposed to work as Nancy and her staff told Mary to just stop by whenever there was a question. But it was a very complex calculation and the business users themselves didn’t really understand it. Mary had just delivered her third redo of the report and the end users were still finding problems with the calculation. Mary came to my office not knowing what to do to get consensus on this one calculation.

I called a meeting with all the stakeholders, including Nancy, her staff, Mary, and me. There were about 12 people in the meeting. I stated that we had been coding the calculation as we had been told, but it seemed that the developers were never told about all of the different scenarios that could affect the calculation. As a result, the calculation was being asked to do different things under many different circumstances. We then asked the business stakeholder to draw out on the whiteboard his understanding of the calculation.

After 20 minutes, we had a whiteboard filled with flows. Everyone participated and there was finally consensus among the group. Yes, they were all ready to sign off on this change. That is, until another stakeholder asked about a condition that was not represented on the whiteboard. Everyone gasped as they realized they had forgotten one specific condition that would affect the calculation about 15% of the time. We then worked that into the whiteboard flow. Again, everyone agreed with the change and was ready to sign off on it. Then, the main business stakeholder, Nancy, who was the most senior person in that meeting and the one who had always resisted process and process improvement, came up with a user workflow that would again totally change how the calculation worked in 50% of the cases! Everyone else had missed it. We added that to the flow on the whiteboard and thought that we finally had the calculation defined.

Mary’s head was spinning, but she did her best to do one final walkthrough with the group. When she was finished, Nancy looked at her, stood up, and said to the group, “I guess this is why process is so important. We should have done this work when the project was starting and I apologize to the group for skimping on the requirements for this. Let me take this back to the business unit for a few days, get a business analyst assigned, and we will then write up a document of exactly what we want. There is no need to spend any more time coding until we do our work.”

I nearly fell off my chair! This was a huge victory. Curt and Nancy used to argue all the time about the need for process, and I knew that this statement from Nancy was a moral victory for Curt, who had left the company by this point. Having Nancy onboard with process was also a moral victory for the entire software development team.

As we walked out of the conference room, Nancy waited for me outside the door. While we were walking to our offices, she said to me, “Ned, it has come to my attention that you allow your group to work from home one day a week. Even on Fridays.” Oh no, I thought. There goes that! I started to mumble something but didn’t know what to say.

“Don’t worry,” Nancy continued, “I just ask that you keep it among yourselves and don’t let the other departments know what you are doing. I see that it works well in your group, but it would never work in mine.”

“Sure,” I said, “no problem. See you later.”

I went into my office and sat down at my desk with a wide grin on my face.