Mystic Microsoft

A Journey of Transformation in the Halls of High Technology

Kraig Brockschmidt

Mystic Microsoft

A Journey of Transformation in the Halls of High Technology

Kraig Brockschmidt

You're invited to copy, print, and share this book…

It's free and it's legal

Mystic Microsoft is published under the Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License (see next page)

This means you may freely and legally share, copy, distribute, and display this book without the need to worry about lawyers, royalties, and all that sort of stuff. This book's website (www.mysticmicrosoft.com) even gives you all the files you need to print and bind your own copies.

Of course, you are not allowed to make any changes to this work, nor are you allowed to use it for commercial purposes or profit from it in any way without permission from the author.

As this book is offered freely, readers are encouraged, though not required, to reciprocate in two ways:

(1) Express gratitude to the author by making a monetary contribution to the author's work and/or writing a positive testimonial about the book with permission to use your words in promotional activities. See www.mysticmicrosoft.com for details or write to the author's address on the next page.

(2) "Pay it forward" by sharing the book with others and/or making a gift of money or volunteered time to a worthy cause of your choice. So that these gifts do not go unnoticed, please inform the author of your gifts via www.mysticmicrosoft.com (or regular mail) so the website can show the positive contributions that this work has inspired.

Mystic Microsoft by Kraig Brockschmidt
First Edition, July 2007

ISBN: 978-0-6151-4379-8

If you have been given a copy of this book in either printed or digital form, you are encouraged to express your gratitude in the spirit of reciprocation described on the previous page. Contributions to the author's work can be made at www.mysticmicrosoft.com or via mail ($5 suggested). To inform the author of other gifts made on behalf of this work, please visit the book's website, send email to gifts@mysticmicrosoft.com, or via regular mail.

Questions about distribution, printing, or usage to can be directed to permissions@mysticmicrosoft.com or the address below.

Visit the book's website for photos and other extras:
www.mysticmicrosoft.com

Autographed copies of this book are available from the author for $20 each (shipping included). Order from the website or send check/money order to:

Kraig Brockschmidt
7410 SW Oleson Road, PMB 389
Portland, OR 97223-7475

(Note: please check the website for current mailing address.)

Mystic Microsoft

Contents

PROLOGUE

A Trend Inverted

It's become increasingly popular in today's business environment to explore the role of spirituality in the workplace: how spiritual principles can be applied to improve one's business and increase employee productivity. Two domains that have long been considered as incompatible as a casino and a convent have found common ground in the drive for success. Corporate leaders, for instance, are finding that honesty, kindness, and generosity are effective business tools. Workers take up a practice like meditation to manage job stress or hone their mental efficiency. Some take up timeless physical disciplines like yoga to firm their bottoms, perhaps at the insistence of employers who are looking to firm their bottom lines. Others pray for guidance in their business decisions or embrace religion—as reported in a recent USA Today cover story about a professional baseball team—to improve their performance on the field. The clever ones even find ways to package and market spirituality as a business in itself!

This is all well and good; there is certainly a place for spirituality in the world of money and success. In fact, it's an ancient practice. Some of the oldest scriptures in the world, the *Vedas* of India, are chock-full of methods to deal with all sorts

of needs, from money and healthy children to power over your enemies and increasing crop yield. The ancient Indian epic, the *Mahabharata,* tells of kings hiring priests to perform rituals on their behalf through which those kings would acquire certain boons or advantages in warfare. Be it victory on the battlefield, Wall Street, or the baseball diamond, the story is the same: spiritual power can be harnessed for material ends. At least when you pray for success, you're more likely to be grateful to God when it comes rather than showering your own ego with self-congratulations. Better to remember God in this way, the authors of the *Vedas* concluded long ago, than to forget him* entirely.

We see, then, that the underlying assumption of the modern trend is that the highest purpose in life is basically to get rich and powerful. Why so? Why are we so caught up in money, power, and success? The answer is simple: we believe that these things will make us happy. We want wealth so we can acquire those things (including relationships) that promise happiness. We want fame so people will love and respect us, which we think will make us happy. We want power and influence so we can control at least some portion of the world, removing conditions we believe cause unhappiness and establishing conditions we believe will, again, make us happy.

Look at everyone around you; look at your own desires and ambitions. Follow the links in the chain to the real end-game.

Any way you slice it, happiness is the secret hunger behind all human striving, the real purpose behind all that we do. Not

---

* I've chosen the masculine pronoun here for simplicity and to keep with common convention. I've also kept such pronouns in lower case, contrary to the usual convention, except where grammar demands. No disrespect or irreverence is intended. It's simply a stylistic choice to keep the text more personal and immediate rather than formal or distant.

just the mere absence of pain or the fleeting satisfactions of sense-pleasures, mind you, nor something static or fragile. We seek an inner state of ever-new delight—a dynamic state of blissful being—that we don't have to constantly defend or buttress against ever-changing threats. For the very fear of loss is what drives us to desire money, power, and influence in the first place; through them we believe we can both acquire happiness and the means to guard and protect it. If we can just grab hold of happiness—*just once*—and make suitable arrangements to maintain it, *then,* perhaps, we'll be at peace in that joy.

Thus it is that we wholeheartedly yoke spirituality and religion, as we do with every other means at our disposal, to the wagon train of material fulfillment. God's grace becomes a commodity, a favor to be won; the Creator someone with whom we negotiate deals; and spiritual practices like prayer, meditation, and right living the secret ingredients to enhance profits and boost the stock price.

Yet there's an insidious irony here. As mystics throughout the ages have declared, the experience of God's presence (however you wish to define it) *is* the very joy we seek, and experiencing that joy is *exactly* what spiritual practices were designed for! Take the Ten Commandments—God did not engrave them on stone tablets for his own convenience or as a (rather heavy) book of law to throw at us in some cosmic trial court. He made them for *our* sake, to help us understand and hopefully avoid those attitudes and behaviors that lead to misery.* Derision, dishonor, stealing, killing, and coveting—these blind us to the joy that God implanted in our souls; reverence, love, generosity, creativity, and contentment, on the other hand, deepen our awareness of that inner bliss.

---

* As Jesus said, "The Sabbath was made for man, not man for the Sabbath."

So to harness spiritual power in a roundabout attempt to find happiness through material growth completely misses the point. It's like having a bushel of grain with which you could easily satisfy your hunger for weeks, yet sell that grain to buy a single slice of bread. It makes much more sense to just eat the grain—to use spiritual practices for their intended purposes and to ask, most of all, how we might harness the opportunities of career and business for our *spiritual* growth.

That's what this book is about.

As you have undoubtedly gathered from the title, the story contained in these pages involves one of the most successful business ventures in recent decades and the very heart of high-tech, corporate multinationalism: Microsoft. I was employed by Microsoft in various capacities for eight and a half years—from March 1988 to November 1996—during which time the company underwent its most important phase of expansion. When I began, Microsoft had six buildings housing about 2,500 employees; its minimal market-share products were hardly given serious consideration by industry pundits. When I left, there were at least *thirty*-six buildings plus countless domestic and international locations housing well over 30,000 employees. By then, Microsoft generally ruled the personal computer software market and got more press than many other Fortune 500 companies combined. Technology, success, money, power…all of these defined much of the Microsoft experience during those years.

I certainly shared in that success, achieving a fair degree of wealth, fame, and influence. Professionally, I made important contributions to some of Microsoft's flagship products, wrote two wildly popular programming books, and became a highly-respected industry expert. On the material side, my wife Kristi and I acquired all the trappings of "the good life" and had

enough investments set aside for quite a bit more.*

All this is a moderately interesting story in itself—I think you'll enjoy the many anecdotes about Microsoft's coming-of-age. What makes it much more fascinating is the added spiritual dimension of my experiences during that era. I won't be saying much, however, about the role that spirituality played in that success. Nor do I have much to share on how I might have brought God and spiritual principles into my work without sacrificing success. Why? Because for most of the time I was at Microsoft I wanted *nothing whatsoever* to do with God or religion!

At the point where this story begins I was very much a skeptic: religion had all but disappeared from my personal consideration. Though raised in a religious household, I found more and more that set liturgies, a creed or two, and spending an hour or so each week sitting in a pew just weren't answering my deepest questions about the universe and my place in it. Never satisfied with smallness of purpose, my mind constantly asked the sorts of questions that don't always go over well with pastors and priests.

So shortly after I started at Microsoft I simply walked away from religion...just ignoring it at first, then working my way through—and basically rejecting—just about every definition or image of God that had ever been presented to me. I saw them as too limiting, too restrictive, or simply an excuse for people to argue. Religion, if nothing else, ought to facilitate a

---

* For the record, I am *not* one of those spend-thrift high-tech millionaires who collect vintage helicopters as a hobby. Though I did effectively retire from Microsoft at age 28 (and became busier than ever!), our net worth at the time of writing is under a million. We live on a modest income from investments that meets the expenses of a focused lifestyle (see Chapter Fifteen) but certainly doesn't lend enough to indulge in opulence.

sense of unity, yet throughout history it's given rise to divisive wars, persecution, social control, and countless other evils (not unlike those we ascribe to modern corporations). Thus my primary interest in "all that religion stuff" was to get beyond it altogether. My energies were wholly focused on my career.

Spiritual growth, however, isn't something we can so easily cast aside. The impulse to expand our awareness in some way is inherent to human nature, inherent to the joy that lies within us. No matter how hard we try to suppress it, that impulse invariably finds some form of expression.

In my case it expressed itself as a desire for truth: I wanted to know how life worked; I wanted to know how everything was connected; I wanted to see the "big picture." Consequently, I devoured a great many books and sought to understand life as best I could. I just didn't want much to do with the "God" thing. I wasn't going to go anywhere near churches or temples or even think of the whole process in religious terms.

Such is the difference between spirituality and religion. Whereas religions are defined by their outer forms, spirituality is strictly a matter of whether one's inner awareness—one's consciousness—is growing and expanding toward the greater reality we call "spirit," irrespective of form. What makes any thought or act "spiritual," including the business of making money, is whether it uplifts you toward that greater reality *from whatever level of consciousness you happen to be.* As such, it's an individual question, not an institutional or social one; actions that uplift a beggar might be degrading to a saint. Similarly, what makes any thought or act "worldly" or anti-spiritual, including anything done in the name of religion, is whether it diminishes your awareness of that greater reality. Spirituality is a matter of *direction*, not definitions. It deals with what works to dynamically uplift consciousness; it has nothing to do with blind dogma, sectarian minutiae, or any

other kind of static belief system (including skepticism) that refuses to test its own validity.

Spirituality is a real concern for each and every human being. While one may or may not choose to participate in formal or organized religion, or even "believe" in anything, every person has some higher potential toward which he or she aspires. Kindness, generosity, honesty, courage, and dozens of other noble qualities are not noble because we, as a society, have agreed upon them as such but because they are expressions of this potential. Customs like marriage are valued not just for their practical benefits (providing a stable environment for children, avoiding sexually-transmitted diseases, etc.) but because soul-qualities like loyalty and commitment are much more in attunement with those aspirations than the superficial "joys" of promiscuity. Indeed, we need only examine the lives of those who actively express higher qualities to see that *they* are the ones who are genuinely happy.

Thus while I thought I could get along just fine by avoiding God and focusing on worldly success, certain spiritual lessons were still necessary for my personal (and even material) growth during that time. The only way I might have avoided those lessons and experiences would have been to completely squelch my desire to grow at all! But if anything I was at least sincere in that desire—I *did* want to grow and expand my experience of life, to whatever degree I understood it. So although I'd basically told God that I wanted nothing more to do with him, he didn't bother to wait for me to come around and commit myself again to religious matters. He simply gave me what I needed exactly where my energies *were* already committed—namely Microsoft.

In short, God used the circumstances and situations of my Microsoft career—success and failure alike—to effect in me a deep, spiritual transformation. In the course of my eight and a

half years with the world's leading software company I learned and experienced exactly what you would expect from direct training in a monastery or ashram: a fresh outlook on the meaning and purpose of life (what you might call genuine faith); a greater ability to remain even-minded and cheerful through adversity; a deeper understanding of universal qualities like patience, perseverance, non-attachment, and simplicity; and the importance of things like good company, selfless service, and receptivity to higher guidance. I also learned and experienced all this despite the fact that for a good part of the time I considered myself an atheist and wasn't even aware I was learning anything!

As improbable as this sounds, the reason is really quite straightforward: the necessary attributes for material and worldly success—namely energy, concentration, and high aspiration, all of which I experienced at Microsoft—are the *exact* same qualities that are also necessary for *spiritual* success. That is why the power of either can be harnessed for the other. The difference, again, is simply one of direction. Spiritual growth is primarily a matter of increasingly directing one's energies toward an expanded awareness and away from selfish, egoic, and materialistic desires. This is the goal of every true religious or spiritual practice: ceremonies, rituals, prayer, meditation, hymns, chanting, and right behavior are all but different ways of raising one's energy and focusing it upward toward Spirit.

As we shall see in this story, an energetic and focused environment like Microsoft can *equally* facilitate this same inner development. Such is the tremendous opportunity afforded to us by our careers. It simply requires an individual dedication to inner growth since most companies themselves are not

spiritually oriented.*

This dedication involves two specific qualities that you will see in the chapters ahead. The first is *sincerity*: having as your underlying motive the search for truth and greater understanding as opposed to seeking only power, wealth, or other forms of personal gain; and asking, in every situation, "what's trying to happen here" rather than "what do *I* want to have happen?" The second quality is *self-offering*: having the willingness to wholeheartedly accept whatever comes to you, good or bad, and to cheerfully (not grimly) commit your best energies to working *through* those circumstances rather than trying to skirt around or run away from them.

Your expression of these two qualities is a way of saying to God, Life, The Universe, or whatever else you want to call it, "I truly want to learn and grow—show me the way!" As a result, God, Life, The Universe—however you want to relate to a greater reality—will respond and guide you, personally and individually *and* in harmony with others concerned, toward your next step upwards. I say this with conviction: if it can happen, as this story shows, within the halls of high technology and without the conscious participation of someone who considered himself an atheist, it can certainly happen to anyone, especially if they *are* more conscious *and* more open!

Thus for those readers who find themselves committed to a career and/or other responsibilities (including family) and who will, for whatever reasons, continue on that course for the

---

* Indeed, a personal dedication is always necessary, even in spiritual organizations. It's actually *more* necessary in a spiritual environment where there's the temptation to think that the environment will do the work for you. People satisfied with their own self-righteousness can go through all the motions for years without actually growing at all. As a great teacher once put it, "It's a blessing to be born into a religion, but a curse to die in one."

foreseeable future, I hope to demonstrate how these things can be an integral, even leading part of a fuller spiritual experience rather than an obstacle. If you give yourself wholly into your duties while holding to your sincere desire to grow and expand, you will find what you need coming to you *within* the context of those same duties—including your workplace. Spiritual and material prosperity can walk hand in hand.

This applies also to younger readers who perhaps feel a certain disparity between taking up an active career of some sort, as the world expects and even demands, and an inner calling to go deeper, spiritually. To you I say that it need not be an either/or question: accepting a career need not compromise one's spiritual aspirations. In fact, I hope this story illustrates how the dynamic and conscious combination of the two can be much more potent—and rewarding!—than fleeing to a remote corner of India or Tibet or dropping out in some other manner.

I also hope that this story will be helpful to those who are making or would like to make a career transition, perhaps to something more serviceful or more directly spiritual. I would help you make the joyful discovery, as I did, of a divine thread running through the tapestry of your past and the deeper purpose of those experiences. With this discovery you can see your schooling and career achievements not as something you're throwing away (as friends and family may challenge you), or as a spiritual waste, but rather as an essential part of who you've become. In this light you can truly honor your past with gratitude for having brought you thus far, then courageously step into a new realm of possibilities.

I'd like to emphasize that the experiences I had, the lessons I learned, and the order in which I learned them were what I personally needed in each phase of the process. The specifics of those experiences and the environment in which I learned my lessons are not particularly important. They're just the back-

drop: don't feel like you have to duplicate them. Whether you're educating children, operating machinery, writing reports, or being on-call 24-hours at a stretch for brain surgery, what matters, again, is your sincerity and self-offering. With these, your unique path will open before you.

Let me also mention that this journey wasn't always easy for me. While there were abundant successes and joys, I certainly had my share of frustration, failure, and even persecution. Nobody said the path was strewn with soft moss and rose petals! But don't expect to see any juicy gossip, dramatic suffering, or bitter finger-pointing within these pages—I'm simply offering an honest account of my experiences.* From the convenient distance of some years I see that both joy and sorrow played necessary and important roles. Thus when I talk of Microsoft, its people, and its leadership, I've made the conscious decision to emphasize the positive. I do this neither to defend them, apologize for any mistakes, or somehow sugar-coat what many people perceive as a big, bad, domineering corporation. I have simply chosen to love the light; let others condemn the darkness. After all, we become what we concentrate on.

That said, this story begins in the fall of 1987, shortly after my nineteenth birthday, when I was just heading out to fulfill all those dreams of worldly success. I had already completed a

---

* While most of the persons involved have allowed me to use their real identities, a few have been changed by request to protect the individuals' privacy. Besides an occasional exaggeration for the sake of humor, that is the only smattering of fiction in this book. I will also add that my experiences were in no way influenced by mind-altering substances, legal or otherwise. I have never done drugs of any kind, I drink no alcohol whatsoever, and have pretty much avoided even caffeinated beverages since high school. If you must know, my biggest vices during my Microsoft years amounted to Twix bars, Grandma's cookies (Double Fudge and Iced Molasses), and caffeine-free Pepsi.

year of college and had, thanks to scholarships and various mundane forms of summer work, no debt and some small savings. My wife and I had also become engaged during the summer with the wedding set for the following July. And now, opportunities to get my career going began to make themselves known.

It was just then that God began his work as well…

CHAPTER ONE

Homecoming

"You should look into the Cooperative Education Program. It's just the thing for a student like yourself."

It was October 1987 and I was visiting an undergraduate advisor at the University of Washington. I had just begun my sophomore year in Computer Engineering and it was time to start looking for relevant summer work.

The University of Washington, among a number of schools, had teamed up with various technology companies to create the Cooperative Education or "Co-op" Program. This was designed to help engineering students—whose experience is, by definition, quite limited—to find some sort of meaningful entry-level work in the industry. The companies created three- to nine-month internships that they would only fill with co-op students. Entry requirements were, of course, kept low, as were the salaries! To a student's mind, though, the pay was *way* better than most other summer options.

The colleges, for their part, would allow students to miss one or two terms without the usual penalties reserved for the academically lazy. At the UW we even got a few course credits to boot. As for the companies, they got to draw on a bountiful pool of eager students who were thrilled to do those "special

projects" that most full-timers find insulting, and were equally thrilled to do it for half the pay and half the benefits. The co-op program also gave these companies an effective way to scout out and even train future employees without having to make any binding commitments in the process.

This arrangement found no argument from me. I made my way to the top floor of Lowe Hall (where the program was administered) and surveyed the list of companies that would be doing on-campus interviews that fall.

I was specifically looking for a place where my computer skills would eventually get me up into orbit. Really. Space exploration was my childhood fascination and I had nurtured dreams of space travel for years. Historically, of course, off-planet adventures were exclusively reserved for crack Navy pilots with perfect vision and entirely closed to only moderately coordinated civilian myopics like myself. But then the Space Shuttle came along and NASA began to toss up "mission specialists" who were needed more for their minds than for their eyes. There was hope!

I came to college, then, to develop those talents of mine that might someday lead to a window seat on the shuttle. As for my chosen major, I first considered mathematics—a subject in which I'd been rather precocious since birth. But early in my freshman year I sat in on the end of a graduate-level math course after which I had a meeting with the professor. For twenty minutes I understood nothing. Zero. Zilch. Nada. I mean it—I didn't understand a single word! What I *did* understand was that I wasn't at all interested in whatever he was talking about. Thus ended any aspiration of following in the footsteps of Leibniz, Gauss, or Poincaré.

I then shifted my thoughts to astronomy which seemed better suited to my purposes anyway. I was particularly attracted to the field of astrophysics not only because it was more tech-

nical but because it also sounded more impressive. The only problem was that finding a job in this field was about as easy as becoming a starting NFL quarterback. Not very promising to someone who was already engaged to be married and talking about houses and families…

That left computers, a field in which opportunities were plentiful and the one in which I already had the most practical experience. My father, you see, had bought me a computer when I was eleven but adamantly refused to buy any software. "That," he told me, "is something you'll have to write yourself." So I did. In high school I even sold some of it. I also wrote articles for a couple of computer magazines and had a regular column in one of them.* By the time I got to college, then, I figured I had the programming end of things pretty well in hand and should learn something about the hardware. Thus I finally settled on Computer Engineering.

As I looked over the list of companies that were scheduling interviews for computer engineers, two of them caught my immediate attention. The first was Boeing, the venerable aerospace pioneer that was taking a leading role in America's space station efforts and also happened to be the career employer of both my father and my father-in-law to be. Certainly a good choice. The second was NASA's Jet Propulsion Laboratories (JPL). I quickly signed up both.

Then there was this young upstart called Microsoft.

---

* The magazines were Rainbow (the largest), Spectrogram (a short-lived, low-budget kind), and CoCo Clipboard (in which I had the column). These focused on the Tandy/Radio Shack Color Computer, a little box with a 1-MHz Motorola 6809 CPU and 64K total memory (K as in kilo- not megabytes). Fiddling with this machine was my primary hobby and my software sales only ever made me enough to buy a new piece of hardware now and then. Nevertheless, it was great fun to share my ideas and creations with others.

Offhand there was little here to interest me. The company was small and its future uncertain; the ink was still somewhat wet on its NASDAQ IPO. All they did was sell floppy disks full of stuff like MS-DOS (yippee!) and this mildly-interesting thing called Microsoft Windows. Sure, it could be fun to work for a small computer company, but as a place to nurture my extra-terrestrial ambitions Microsoft left something to be desired.

I signed up for an interview anyway. I'm not really sure why. There was just this little sense of attraction toward the company, a little inner nudge that said, "why not?" Besides, it just felt better for some reason to have three interviews lined up instead of only two…

My interviews began a couple of weeks later. The first, with Boeing, was a very calm and cordial affair as one would expect from an established institution. I did well answering all those questions about why I had chosen my particular degree and so forth, and left the room feeling confident that an offer would be forthcoming. All I had to do was wait for their call and my orbit-bound career would be launched, so to speak.

My little chat with Microsoft was scheduled for the following morning. I actually thought about giving it a miss since Boeing's pending offer would downgrade my interest in the small software firm from "slimly marginal" to "wholly superfluous." But I figured I might as well go through with it just in case something unexpected came up. No harm either with getting a little more interviewing experience.

Well, something unexpected did come up: I was offered a job before I even sat down! Bob Taniguchi, the man who greeted me, simply said "Good to meet you. I'm happy you'll be working for me this spring."

Say what?

Giving me no chance at all to think about what he had just said, Bob galloped off into what felt like a first day's orientation session rather than an interview. He fired me up (though we were seated now) for working in his Developer Support Group where I would learn so much about programming Microsoft Windows that I could help outside software engineers tackle their most daunting problems. He then painted a vivid picture about rubbing elbows with all the great people at Microsoft* and highlighted all the special perks that "we employees" enjoyed, including the free T-shirts and soft drinks. Then to wrap everything up (after a few obligatory technical questions), Bob flat-out offered me the job again. "I'm looking forward," he said, "to working with you next spring."

As you might expect, I was quite surprised by this rather unorthodox recruiting method. I was even more surprised by my *response* to it all! Instead of writing off Bob as some slicked-over marketing weasel making a low-rung job in some new-kid-on-the-block company sound glamorous—as my cynical nature of the time should have demanded—I had absorbed everything he said like the proverbial sponge. Scarcely five minutes into our half-hour session I felt as if I had rediscovered a long-forgotten family. Everything Bob described about Microsoft and its people resonated with me on some deep level. Something was just so very *right* about all this; my whole being thrilled in a way I'd seldom felt before. And if my answers to Boeing's questions were fairly well in tune with that firm, my answers to Bob's questions—when he finally bothered to ask them—were exacting.

---

* As a Microsoft recruiting brochure of the time put it, "If you want to know something about MS-DOS or Microsoft Word, just walk down the hall: the people who wrote it are probably there!"

I learned later, when asked to conduct interviews myself, that this was somewhat typical of the Microsoft screening process. We didn't necessarily care about your career goals nor did we care all that much about any specific job experience. What we wanted to know, more than anything, was how well you "fit"—in a kind of vibrational way—with Microsoft's unique corporate culture. To that end, we threw you all kinds of challenges, surprises, and apparently insoluble technical problems just to see how you would respond. This told us, with a fair degree of accuracy, what would happen when you were exposed to the intensity of The Microsoft Way.

In my case I don't think there was any doubt. Both my outer and inner responses to Bob's presentation proved that I was true Microsoft Material.

Back then, at least, when Microsoft saw something it wanted, whether it was an individual or an entire company, it went right after it. This was due, I think, to the fact that decision-making power for this sort of thing (during my time there) was usually given to whomever had the most riding on the acquisition in question. A vice-president, for example, could go out and buy another company without even notifying the president or CEO; after all, it was his or her division that had to absorb the costs. As for hiring new employees, that power was pretty much given to the person's would-be manager who could often make a decision on the spot.

As a result, hirings sometimes happened with dizzying immediacy. In early 1992, for instance, one of Microsoft's primary competitors fell on hard times and eventually had to send out the pink slips. Sixteen hours later (as the story goes), the company was horrified to discover that—OOPS!—they'd accidentally canned one of their top software architects. They immediately called him to apologize and make amends, but in that small window of time Microsoft's programming languages

group somehow tracked down this newly available "free-agent" and signed him. Indeed, when his now-former employer called he was already packing for the move!

In Bob's eyes I must have been similarly attractive: the official job offer came at eight-thirty the next morning, only twenty-one hours after my interview. (I can't be too proud—if I had been *really* hot they would've called the same day.)

I was, of course, ecstatic to get my first real, honest-to-God offer, especially one with so much energy around it. But when I was only given *forty-eight hours* to say yes or no, I plunged into inner turmoil. I didn't want to just jump at the first thing that came my way. I wanted to see what Boeing had to offer. I wanted to see what kind of work I might find at JPL. And I still wasn't quite sure about this adolescent software company that had nothing whatsoever to do with my astronautical fantasies. Would Microsoft really give me the experience I needed? Would that experience be valued by other future employers? Was Microsoft even a good *short-term* prospect? Or were they destined to go the way of so many other software startups that had a nasty tendency (well before the dot-com bust) to fall into bankrupt obscurity?

I was horribly confused, even terrified. The universe was inviting me to take a step I didn't really understand at all. I knew I was standing on the brink of a decision that would affect the entire direction of my life. "What should I do? What should I do?" My thoughts kept swinging like a pendulum between rationality and the full gamut of emotions. For every good reason that came to mind for choosing one way or the other I was mercilessly besieged by the forces of attachment, fear, insecurity, worry, and yes, even excitement!

I desperately wanted more time. I wanted time to sift my way through every possibility. But of course, I didn't get that luxury. There must be a universal law somewhere that says the

amount of time you get to make a decision is inversely proportional to its importance. We typically get months to select just the right towels to match the tile highlights in the master bathroom, whereas we only get a few days to choose between two life-paths that lead to radically different destinations. As General Dwight D. Eisenhower put it (reported in *Eisenhower and Churchill* by James C. Humes), "A meeting whose main item was corner windows for heads of departments took almost five hours where the decision on D-Day five minutes."

Fortunately for me, I couldn't just sit there and churn on it: I had my usual classes to attend, homework to complete, and a paper or two to write. So I just had to let it go for a while. After all, I did have forty-eight *hours*, not forty-eight minutes! Plus, I told myself, working for any of the three companies would both help my budding career and certainly be great fun. What mattered, then, where I ended up? Eventually I found myself able to calmly accept whatever outcome was waiting for me.

This was the best thing I could have possibly done. Pulling away from both emotional and rational extremes of the pendulum and just giving myself into whatever possibilities awaited me, I found myself resting—pretty much by accident!—at that one point in the very center where motion ceases entirely. In that stillness, where the inner guidance of soul intuition has a chance to speak, I absolutely knew that choosing Microsoft was the right thing to do. I couldn't have told you *why* it was right, I just knew that it was. Microsoft was where I belonged, let come what will.

The next morning I called Microsoft to accept the offer and cancelled my interview with JPL. I was that certain.*

---

* I did get an offer from Boeing, by the way, two weeks after I had accepted the position at Microsoft.

The following spring I thus entered the halls of Microsoft for the very first time. As if celebrating this new beginning, it just happened to fall on the Vernal Equinox: March 21st, 1988.

After a few hours of entertaining company orientation and all that not-so-entertaining legal paperwork, I met up once again with Bob Taniguchi. Wasting no time, he immediately showed me my new desk in one corner of a double-size office in Building Six, already home to three other co-op students who were busy answering technical support calls. On top of my desk squatted a 10-megahertz "286" computer (a true boat-anchor by today's standards) along with my personal copy of Windows (version 2, for those who remember it), all the necessary programming tools for the system, and a book called *Programming Windows* by one Charles Petzold. Though I was utterly thrilled, I was also a little nervous: I had worked on an IBM-compatible computer only briefly, I knew next to nothing about Windows, and I had never even heard of the particular programming language (called "C") that I now needed to learn. Bob was fully aware of these shortcomings. Yet in the truest tradition of The Microsoft Way he simply said, "You're on the phone in two weeks!"

That being the case, I dove into my work wholeheartedly. Living alone in an apartment less than a mile from Microsoft (our wedding wasn't until July), I spent each night devouring books and programming manuals until sleep won over. At work during the day I wrote experimental programs and listened in on other support calls. And two weeks later, when I was figuratively kicked out of the nest, I managed to fly pretty well on my own. In fact, it wasn't long before I was truly enjoying every day's work more than I thought possible, so much so that from my first day on I never once thought about the opportunities I might have missed elsewhere. Whatever dreams I had once nurtured vanished in the deep inner knowing that I was exact-

ly where I *belonged*, where I would find exactly what I needed regardless of what I thought I wanted.

For like a newborn child, I had come home into the family with whom I would share my next phase of growth. And here I would stay and serve until Microsoft had fulfilled its purpose in my life.

CHAPTER TWO

Baby Steps

"First comes the test of fire.
Then comes the test of ice.
Then comes the test of patience."
–Ken Cohen

"Writing a Windows application is like having Microsoft give you a periodic table of the elements and asking you to make a broccoli."

This infamous remark of writer and consultant Alan Cooper, who is honored as the father of Visual Basic (Microsoft's most revolutionary programming tool), pretty much said it. Back in 1988, before anyone started making all the powerful tools that programmers enjoy today, writing an application program or "app" that ran on Microsoft Windows was complicated and confusing. Most programmers, being under the pressure of some arbitrary deadline, had little time to really learn the system before diving in and trying to get their projects off the ground. That's why they called on us in Microsoft's Developer Support Group.

I had a deadline of my own. As mentioned at the end of the last chapter, I was given only two weeks from the day I started

before I was on the phones. Fortunately I picked up the periodic table quite quickly and made my first "broccoli" in the somewhat short span of three weeks. It was a little program that drew an assortment of interesting spirals using lines, circles, diamonds, and, my mother's favorite, the silhouette of a Scottish Terrier. While this app wasn't particularly useful and is the kind of thing you can write in a couple hours with today's tools, it was a major accomplishment given the circumstances. I had every reason to be very proud of myself—I had learned to walk quickly enough that, proportionally speaking, it should only be a matter of weeks before I would learn to run, fly, and sail to the stars!

Goaded by my success, I soon conjured up a second project that would actually produce something useful. As an engineering student I had to work through an endless stream of computational problems. For these I had always employed my trusty Dynatone scientific calculator, the one for which K-Mart grossed $10 during my first year in high school. Cheap, to be sure, but it was far more helpful than the small calculator program that came with Windows at the time: an on-screen rendition of one of those chintzy four-function jobs that you get free with a completed credit card application or a paid subscription to some consumer magazine. Woefully inadequate for my needs. So using my precious Dynatone as the model, I set out to create a full-blown scientific calculator for Windows.

Success! In again only three weeks, between the support calls I now routinely handled, I completed my new project. It had all kinds of interesting features. It would of course add, subtract, multiply, and divide like the existing program, but this one would also do factorials, logarithms, and hyperbolic cosines to twelve significant digits! It could perform statistical calculations with piles of data and could do logical operations in binary, octal, and hexadecimal number systems—a must for

computer engineers. And it even allowed you to "paste" in a series of operations written out in text, which it would then work through as if you had keyed it all in manually!

My little creation was a smash hit among my full-time co-workers, the ones whom us student intern types looked up to with a certain degree of awe. Every day one or more of them would thank me for what was proving to be a very valuable little program. And Bob Taniguchi, my group manager, liked it so much that he proudly showed it to the development team working on the next release of Windows, version 3.0.

Success again! The Windows team dumped the old calculator in favor of mine. "Yippee!" I cheered! I was thrilled. To have *any* piece of your code included in a commercial software product—well, that was the greatest achievement that any college intern could hope for, especially for a humble little co-op student in Product Support. And to have your work become so *visible* in your company's flagship product? Wow!

Not only that, but Bob successfully lobbied the Windows team to display my name in Calculator's "About…" box. Boy, was I proud. Hall of Fame, ho![*]

I imagine that at this point my more experienced associates were shaking their heads in amusement. Here I was, imagining

---

[*] The Windows team agreed to this because I wrote the program on company time and thus wasn't eligible for any other kind of bonus. Bob also promoted the idea that such visible credit would be helpful when I (supposedly) interviewed with other companies after graduating from college. Thus the credit line "Developed for Microsoft by Kraig Brockschmidt" first appeared in the Calculator of Windows 3.0 (see photo on www.mysticmicrosoft.com) and—to my continued amusement—in every copy of Windows and Windows NT until 1998, numbering well over 100 million. The credit only disappeared in Windows 98 after I'd left Microsoft, while the program, without the credit, continues to be included with every copy of Windows at least through Windows Vista. Having my name appear for as long as it did was an incredible run especially considering that I hadn't touched the program since the summer of 1989. If nothing else, it offers a fun way to introduce myself.

myself a highlight in the Who's Who of Programming while yet wholly ignorant of the hard reality to come, namely that every program needs considerable refinement before it reaches maturity. And in my excitement I didn't bother to notice that my contribution was about as important to Windows as a small concession stand is to a major-league ballpark.

My ignorance was soon remedied. After the Windows team added my code to their project it became subject to the rigorous testing (no joke!) that Microsoft applies to all its products.

Microsoft development groups have three basic positions: program manager, software engineer, and tester. The program managers are the ones who try to dream up things that customers can't possibly live without and are therefore willing to buy in quantity. Their job is to write the product specifications. These "specs" are then passed along to the software engineers whose job it is to manifest those specs in a working program. That program is then passed on to the testers. And *their* job is to mercilessly abuse the tar out of the thing to see whether it lives up to the spec.

As a programmer, you are supposed to love your test team because they're critical to the production of decent software. But you really hate them because they're usually just too damned good at it! If, for example, you give them a shiny new car that you truly believe meets the indestructibility standards of a military transport, they will dutifully drive it through a suitable war zone. Then they'll hand you the wreckage with some really helpful comment like "it broke." *You* as the programmer get to figure out *why* it broke, and you get to figure out how to fix it. Then once you think you've got it figured out, you have to let those pitiless thugs thrash on it all over again. Only when *they're* satisfied with your work is it deemed ready for the kinds of abuse that customers will inflict on it, which is, of course, far worse.

My first version of the new Calculator was about as solid as a car that expected to be gently rolled along a dead flat, newly-paved road at about 4 mph. The testers, in other words, were soon sending me shrapnel. At first I balked at the idea that there could be anything at all wrong with "my baby," but then learned to accept the fact that they were actually helping me fine-tune it.

"OK," I said to myself, "if that's the goal then let's really do it right!" I resolved to do my utmost to create the ideal program, as small and efficient as possible with absolutely "zero defects," and throughout the rest of my six-month internship I happily corrected any errors that the testers discovered. Then when I returned to school that fall, the Windows team offered to make me an "unofficial" contractor so I could keep in touch with them and continue bullet-proofing my program. This really only meant that I got to retain my Microsoft email account: I didn't get a penny for my work. But that didn't bother me—the sheer glory of having my Calculator included with Windows was enough of a reward.

Now while I learned to appreciate the testers, I can't say the same about the program managers that began weaseling into the picture. They kept asking me to make "improvements" to Calculator that seemed totally ridiculous: modify the user interface, change this color, add this feature, remove that function…. To make matters worse, a number of times I'd make a change only to be asked to remove it a week or two later. It started to drive me nuts! Wasn't my program already designed as well as it could be? How could they have the nerve to change something that was already so *clearly* perfect? Why were they asking me to do things to my program that *obviously* didn't have anything to do with its efficiency? In my youthful ignorance I just couldn't understand what the hell they were trying to accomplish. I even got really irritated when their requests

caused the size of Calculator's executable file to deviate from a precise 40,000 bytes!

I hope that by now I've communicated the nature of my own arrogance in the whole matter. I didn't like the way things kept changing, especially after I'd committed myself emotionally to one way of doing things. Attachment is a sure-fire recipe for frustration.

What I failed to understand was a somewhat unique aspect of Microsoft's product development cycle. In other companies, so I've heard, specifications are actually finalized *before* the programmers start writing any code at all. Not so at Microsoft: in the dynamic world of personal computer software, every member of the product team works simultaneously. Program managers, in particular, are constantly adjusting a product design according to changes in the marketplace or the simple feasibility of implementation. As a consequence, they keep on changing the specs and the software engineers have to keep changing the code: the specs, in fact, are not considered final until the day the product itself goes to manufacturing!

In order for this rather fluid arrangement to work at all, it is vital that a product has an overarching vision or ideal to guide it. The primary focus of Windows 3.0, for instance, was ease-of-use, itself only one facet of Microsoft's overall corporate mission: to improve quality of life through personal computer technology.

You might notice that unlike many corporate mission statements about becoming "the market leader in non-chlorine toilet bowl cleaners" and the like, these abstract ideals like "ease-of-use" and "quality of life" say nothing about Microsoft's own success. This stems from the fact that Bill Gates, contrary to popular opinion, did not create Microsoft out of a desire for personal gain or glory but from a sincere desire to share the joy of personal computers with everyone. He rightly assumed that

success would naturally follow.*

With this high ideal enshrined at its very heart, Microsoft (in my experience) continually challenged its product development teams to operate on a scale that transcended their own goals as well as those of any individual. Employees were encouraged to maintain an expansive outlook in their work, seeing it in terms of offering something of real value to the world rather than merely making money. In this way, corrosive office politics and interpersonal rivalries were rare. Managers seldom had to give pep-talks or sermons on teamwork and just about everyone was willing to put out a little extra effort when necessary.†

So while I thought I'd found my entry into programming history with one spectacular leap, the truth is that these first baby steps of mine, wonderful as they were, merely brought me to the base of a steep mountain. And if I was to climb that mountain—that is, if my little Calculator was ever going to see daylight—I would have to let go my own personal "ideals" and embrace the broader vision of Windows 3.0.

With my desire to have my name in lights (or at least on-screen) being stronger than my attachment to particular colors, features, or the size of Calculator's executable, I gradually

---

* From what I saw of Bill, this motive runs far deeper in him than any thought of personal reward. Otherwise I imagine he would have jumped ship years ago rather than endure the persecution to which both he and Microsoft have been subjected.

† Of course, like everything else in our imperfect world, Microsoft was not entirely successful in its expression of this ideal. "Extra effort" was sometimes taken for granted or even made a requirement; personal desires did flare up from time to time, especially when Microsoft stock wasn't performing like people thought it should. But by and large these were the exceptions and not the rule.

became willing to do whatever the program managers asked of me. And by focusing more and more on the higher purpose of the overall Windows product, my own whims and fancies fell away. In fact, I eventually forgot about my credit line in Calculator's "About…" box altogether.

This is exactly the purpose behind all high ideals: they help us forget ourselves and our personal concerns and embrace a larger reality. Growth of any kind cannot happen without some kind of expansive vision. We must have something toward which to grow if we are to grow at all. As Voltaire put it, "If God did not exist, man would find it necessary to invent him." At the same time, we shouldn't get carried away with lofty thoughts and lose sight of the fact that growth—whether technological or spiritual—is a step-by-step process, never a sudden change. Patience, it has been well said, is the fastest route to God—or any other grand aspiration.

Observe, for instance, how NASA successfully put men on the moon and returned them to earth months ahead of President Kennedy's so-called "impossible" deadline. Starting with next to nothing, NASA engineers first learned (with Project Mercury) the basics of space navigation using simple one-man capsules attached to the top of existing Army rockets. From there, and despite many failures, they designed the more ambitious Gemini projects through which they learned complicated maneuvers and refined longer-term life support systems. Then with Project Apollo they learned how to launch much greater payloads into orbit and to send a capsule around the moon. With all the pieces in place they were finally ready to take one small step for a man and consummate that one giant leap for mankind.

Similarly, one of Microsoft's greatest strengths has been the willingness to work toward an ideal product in distinct stages, putting off certain features for many years until the develop-

ment team is ready to implement them and the market is ready to accept them. Consider Microsoft Windows itself, which spent several years in development before first hitting the streets in 1985. Well, "hit" isn't quite the word—version 1 hardly drew a glance from the public eye. In the broader context, however, it was the necessary foundation for the much-improved Windows 2.0, released in 1987. And while version 2 still failed to gain widespread popularity, it paved the way for Windows 3.0, released in May of 1990. This version finally caught on and began the Windows revolution. Even so, the product hadn't yet reached the designers' original vision: it took another eight years to really get there through the releases of Windows 3.1, Windows 95, and Windows 98. And in that time, of course, the vision itself continued to expand, as seen in the more recent incarnations of Windows that are themselves intended to set the stage for even better things in the future.

Within an ever-expanding reality like this, its important to only add new features when the time is right and no sooner. Otherwise a project (including one's inner growth) just gets way, way out of hand. There was once an ambitious database project called Omega, for instance, that was as glorious as it was impossible to actually build. What we know today as Microsoft Access was resurrected from the ashes of that fiasco. Microsoft Exchange and Outlook similarly grew from the remnants of an idealistic do-everything-in-the-workgroup-universe product code-named Laser. And the highly popular programming tool called the Microsoft Foundation Classes (MFC) sprouted on the grave of an earlier design that, though thought to be absolutely perfect, had to be trashed *in toto* when its initial implementation virtually imploded.

I consider it an act of grace that I fell prey to this sort of impractical rapid-expansion spirit at a point in my career when I could only cause minimal damage. But I did my best! It was

the spring of 1989 and I had been working with the Windows team on Calculator for nearly nine months now. By this time I had fully embraced the higher ideals of Windows 3.0 and had even proved myself to be a reasonably competent software engineer. For these reasons I was offered an official (that is, real money) contractor's job to do some needed work on the other small accessory programs that came with Windows, such as Notepad and the since-retired Cardfile, Calendar, and Clock.

One of my tasks was to incorporate a new digital font into the Clock program to make it look, well, more digital. The font given to me for this purpose was nice but the numbers seemed a bit stiff—they were all straight up and down, like 12:35, not slanted slightly to the right, like *12:35*, as one usually saw on a digital LED clock. So I didn't just add the new font, I went a step further and italicized it. Sure, it wasn't part of the spec, but what harm could there be in it? Surely everyone else would like my little aesthetic adjustment—after all, we were doing everything we could to improve Windows 3.0, right?

Now within Microsoft development teams, people generally don't walk around all that much and talk face to face. On large projects, especially, a great deal of communication happens through email and through some kind of project-management software. At the time, the latter was a tool called RAID (as in the bug spray) that primarily maintained a huge categorized list of feature requests and known program bugs.* Every entry

* In programming jargon a "bug" refers to some flaw in a program that makes it malfunction or produce incorrect results. Computer folklore has it that the first such use of the term came from the era when computers were built with vacuum-tubes and mechanical switches rather than solid-state transistors. Apparently a moth got into one of these computers—either the Harvard Mark I or the Army/University of Pennsylvania ENIAC—and got squished inside a relay. As moths are generally not electrically conductive, the relay didn't make contact like it was supposed to. Thus it was a literal bug (though entomologically a moth is not an insect) that caused an error in the program.

in the list described the problem or request, its entire history, who was responsible for dealing with it, and its relative priority in the project as a whole.

Whenever program managers wanted you to add a feature, they created a new entry in RAID and assigned it accordingly. Testers did the same when their ruthless throttling invariably revealed problems in your code. As a software engineer, then, your personal "to do" list were those RAID entries currently assigned to you. Every day you'd look over your list and work on whichever ones had the highest priority. When you finished adding a feature or correcting some problem, you marked the appropriate entry as "resolved" and assigned it to the test team for verification. If they were fully satisfied, they marked it as "closed" and all was well. Otherwise they'd "activate" it again with some comment about what didn't work, and the whole process started over. Sometimes things had to go back to the program managers to be redesigned; occasionally a single bug would cycle around the chain several dozen times![*]

Between email and RAID, then, you could get all kinds of work done without ever having to talk directly to another human being. This was helpful when you consider that everyone in a Microsoft development team was generally free to come to

---

And the fact that it took an annoyingly long time to discover the real source of the problem set a bothersome precedent that has remained in effect ever since.

[*] A friend of mine who worked as a tester on Microsoft Excel once logged a bug against the cartons of milk in the free drink coolers. He noticed that the chocolate milk failed to list the ingredient "cocoa" whereas the plain 2% milk did. An intense discussion in RAID over the relative merits of these "features" continued for three or four weeks and involved as many as a fifteen different engineers. I think it set some kind of record. Anyway, someone finally went so far as to notify the dairy itself and the bug was closed as "WON'T FIX—ASSIGNED TO VENDOR."

work whenever they wanted, day or night, or, oftentimes, day *and* night. During "crunch" mode, especially, you could usually find someone actively working and someone actively sleeping at any hour of the day on any day of the week. As for myself, I did most of my work during evenings and weekends since I was still going to school in the daytime.

So I had italicized the clock font with the hope that people would see it and *like* it, perhaps even drop me a compliment or two for my creativity. The only response I got, however, was a new RAID entry assigned to me in which some tester, whom I didn't know and had never met, stated with heartless indifference, "It's not in the spec…remove it."

Well! I'd already become somewhat proud and attached to my special feature and wasn't going to be put off that easily! So instead of removing the italic font I simply made it optional: you could toggle italics on and off by pressing "Control-I." Problem solved! I resolved the bug in RAID with a glowing report of my latest brilliant innovation. Everyone else would surely accept my work now.

No such luck. Little features like mine, no matter how innocent—or even useful!—were considered a serious liability to the overall project. Features needed to be tested. They needed to be documented. They made the project unnecessarily late and unnecessarily bloated. And on a project that was already late and already bloated, the project managers—known as the "code police" by the more renegade programmers—were hell-bent on keeping out any and all superfluities. This included italicized clock fonts as much as it did the hidden bits in one of the system's core modules that displayed, when you issued the secret command, a Klingon battleship with the slogan "GO AHEAD, MAKE MY DAY." If such needless waste was allowed in the final system it might all add up to several more floppy disks in the product box (we didn't have CD-ROMs let

alone DVDs yet). This would raise the cost-of-goods per unit. This would cut into net profits. This would lower Microsoft's stock price. This would reduce the value of everyone's stock options. This was a ***cardinal sin***.

Well, as a lowly contractor and member of They Without Stock Options, I didn't know about any of this—nor did anyone bother to educate me. In fact, this time I wasn't even told to remove the italics: someone else did it for me!

When I discovered this latest snubbing of my "genius," I became truly rebellious. "How dare they!" I cried. "I'll show them!" I shrewdly added my feature back in again without noting the change in RAID. If they didn't know they wouldn't care, right?

Wrong…my change was still obvious. In a big development group, there has to be some kind of control system built around a project's source code (all that weird-looking symbolic stuff that only programmers understand). Without such a system, programmers would overwrite, undo, or erase each other's work without even knowing it. This, as Microsoft discovered years earlier, is very, very bad.

So we had another set of tools called the Source Library Manager, or SLM, to coordinate code changes. SLM, which we affectionately pronounced *slime*, maintained a single copy of a project's source code on a central network server.* To become part of the project you first "enlisted" in it. After that you could check out individual files to take exclusive control over them.

---

* Recently I found this gem inside some obscure Microsoft documentation:

slime (slīm) *noun*
A thick, sticky, slippery substance.
A mucous substance secreted by certain animals, such as fish or slugs.
Vile or disgusting matter.
Source Library Manager

This allowed you to make modifications with the assurance that no one else would be doing so. When you finished your modifications, you checked those files back in. This merged them into the central copy of the source code and once again made them available for others to check out. This much I understood.

What I didn't know about SLM is that every modification to a source file was automatically recorded in a change log along with your email name. What's more, everyone enlisted in the project was notified of the event!

Needless to say, the Software Inquisition watched these notifications like psychopathic snipers. They saw my change and took it out again without comment. I still have no idea who these people were, but they were sharp.

Stubborn as I was, I still didn't admit defeat and resorted to an even more subversive tactic. If you had sufficient mischievous intent you could readily access the central SLM computer and directly modify source code files without checking them out, without checking them in, and without creating any entries in the change log.

I was sufficiently mischievous.

Once again I silently made my change.

And once again it was silently discovered and silently removed.

To my lasting astonishment, no one told me about it, no one asked about it, I wasn't held accountable for it, nor was I disciplined in any way. Perhaps I was laughed at behind my back, but I only heard silence. So with fortune still on my side, I finally (and wisely!) gave into the truth that my great inspiration simply wasn't going to happen: reckless idealism had met final defeat at the hands of practicality.

Only later did I finally learn that the only reason for the digital font in the first place was so Microsoft could say that

each and every one of the accessory programs in Windows 3.0 had been in some way improved. For Clock, this one change of the font was sufficient—no more, no less. It was the one and only step needed at the time; other steps would be taken later, when appropriate.

So ended my period of youthful idealism. "To every thing there is a season and a time to every purpose under the heaven." This famous Biblical dictum held true in the case of Clock. Microsoft's next step for it in Windows 3.1 *did* allow you to italicize the font. In fact, you could actually choose any font you wanted and make it italic, bold, or underlined. This was a much better solution than mine but was too ambitious for Windows 3.0. So I feel somewhat vindicated that my idea was at least going in the right direction. In any case, the whole point is now moot—since Windows 95 the clock has only been a tiny speck on the "task bar" where the font is far too small to even be an issue.

Looking back, it's clear to me now that through all my foolishness—but sincere foolishness—with Calculator and Clock, I was simultaneously being taught valuable lessons about the importance of idealism itself and the importance of being patient and *practical* in that idealism. Without these it's altogether too easy to lose perspective. Without direction, your steps fail to produce meaningful and lasting change, and you lose hope. When the steps are too wide, the path before you begins to appear increasing difficult and you gradually convince yourself that high ideals are wholly unattainable. Either way you simply stop growing.

Few lessons in life are more critical than this one. As I said earlier, it was an act of grace that it came at a time when the risks were so minimal. Coming when it did, it taught me an approach that has simplified my daily life quite considerably: Yes, let your ideals inspire you to the highest you can imagine,

but let them also inspire you to find ways to make them real and meaningful, right here, right now. Be willing to put things off until you are ready for them; work toward your ideals with patience. For by applying them to your present reality they will harmoniously lead you, one step at a time, toward that which you seek.

It is, after all, how we put men on the moon.

And it's how each one of us can reach for the stars.

CHAPTER THREE

Pole Shift

"Only one in ten students will ever hold the position of 'design engineer.' "

My favorite computer engineering professor, Dr. Yongmin Kim, was addressing my class in digital circuit design with this sobering statistic. His purpose was to help us be realistic about our chosen profession: while every one of us enjoyed the work of creating new computer programs and circuit designs, only a few of us would ever get to do it in our full-time jobs.

Dr. Kim also helped us understand how demanding the work could really be: his notorious senior-year design courses were intentionally difficult. The words "severe" and "oppressive" often arose in their context. At the same time, those who really gave themselves into their projects learned a great deal about computer technology—and about themselves.

At the University of Washington, engineering hopefuls can't actually declare their intended major until their junior year. Instead, you spend your first two years as a pre-engineering student in the College of Arts and Sciences, taking all kinds of

introductory engineering courses. Only when you have fulfilled these requirements are you allowed to apply for entry into a specific department of the College of Engineering. If admitted, you then declare your engineering major; if not, you continue in the College of Arts and Sciences with a declared major in something else like Physics, Chemistry, or Mathematics.

Of course, each engineering department has fairly high admission standards for the simple reason that the quality of their graduates determines the department's reputation and, to a very real extent, the size of their research grants. It's in their best interest to be picky and the Computer Engineering Department was no exception.

By the time we shared a classroom with professors like Dr. Kim, then, we had already come a long way. As a result, and despite his exhortations to the contrary, I think every one of us hoped to beat the odds. Doing the really important, *creative* work that goes into world-class technology was the greatest glory we could imagine for ourselves—none of us aspired to the relatively plentiful but presumably wretched jobs of testing, documentation, product support, and–God help us!–marketing. Yuuuugh…

I was especially determined. Throughout my upbringing it was expected that I would do important things—you know, become a multi-millionaire, win a Nobel Prize, make some world-changing discovery, that sort of thing. It was only natural: I had always been at or near the top of my class, I always got high scores on various aptitude and achievement tests, and one IQ assessment even pegged me at 160.* I was, in short, a

* I have little faith in the results of IQ tests; some of the most so-called "brilliant" people are also some of the most unbalanced. There's much more to being a successful person than scoring well on what are clearly lopsided tests.

hot-shot. I believed it, my parents believed it, and so did plenty of others. Therefore I wasn't interested in Dr. Kim's warnings: *I* would be one of the elite. In fact, wasn't this already a foregone conclusion? After all, my Calculator program was going into the flagship product of what was rapidly becoming the *crème de la crème* of software companies! All I had to do was finish this little formality called "college" and my career as a design engineer would become a reality.

Toward this end, Bob Taniguchi suggested that I interview with Microsoft for a "real" summer internship. This was the next logical step for me. Unlike co-op students who only did the rather pedestrian work of answering the phones in Product Support, *real* interns did *real* software engineering on *real* Microsoft products, stuff that was usually far more critical to its success than something like my Calculator. For this reason it was fairly easy for successful interns to get hired on permanently once they'd finished their studies.

Of course, this meant that Microsoft screened interns as stringently as they did full-timers. Like my engineering department, Microsoft was looking for hard-core applicants who were willing to undertake enormous challenges and full professional responsibility. And in those days especially, when Microsoft was trying to gain its first footholds in many different markets, they were very serious about hiring only the best and brightest. Thus was developed the dreaded Microsoft Interview to find them.

Many companies, so I heard from fellow classmates, only spend an hour or two with full-time candidates asking about strengths, weaknesses, career goals, and other miscellaneous drivel. Sure, Microsoft occasionally asks some of these same things, but only as a prelude to as many as *eight continuous hours* of intense scrutiny. Hour after hour you are challenged with astoundingly difficult and unorthodox technical problems,

devised over many years by the most creatively sinister minds in the company and carefully guarded as strategic corporate secrets. Under such pressure, your Inquisitors can observe how quickly you comprehend intricate procedures and can exhaustively probe your ability to solve even an apparently simple problem under a variety of real-world constraints.*

To make it even more fun, each hour-long segment of the interview builds on the sessions that precede it. As soon you finish one session your interviewer takes you to the lobby of the building in which your next interrogation will take place. While you have a few minutes to engage in some serious self-doubt about your performance thus far, he or she then pays a visit to the office of your next exploratory surgeon to plot your further dissection. Then after you are safely escorted to the operating room, so to speak, the person who just finished giving you a once- or twice-over writes up a detailed evaluation and emails it to everyone else on the schedule. Knowing thus what has already been excavated in full, every subsequent session goes even deeper into the marrow. In short, nobody needs to *ask* about your strengths and weakness—The Microsoft Interview makes them all too apparent!

My own internship interview in November 1988 went on like this for an entire day; it made the worst of my final exams in college seem like a pop quiz. I was, so to speak, grilled, charbroiled, fried, and roasted one hour to the next.

---

* Examples of such conditions include having (a) a computer with no memory but a very fast microprocessor, (b) a computer with a really slow processor but gobs of memory, and (c) a slow computer with very little memory but a nearly infinite hard drive, combined with the goal of (1) make the program as small as possible, (2) make the program as fast as possible, or (3) get the program working as soon as possible. An interesting book on similar but non-technical aspects of Microsoft interviews is *How Would You Move Mount Fuji?* by William Poundstone.

This was literally true during lunch: my technoanalyst for that hour first asked me if I liked ethnic food.

"Well…I haven't really eaten much of it," I replied, remembering the times I'd ordered American when my family went out for Chinese.

Ah ha! There was his opening. "How about Thai food?" he asked.

"Sounds fine to me!" I agreed, shakily. As much as I was a culinary xenophobe, how could I say otherwise?

We went to a restaurant that must have been called *Fire of Siam* or something like that. Correctly assuming I had never eaten Thai food before, my host issued a subtle challenge as we sat down. "I have a friend," he said, "who's working up the spicy scale to 'five stars.' I think he's gotten to four…" Well, with a comment like that I *had* to order a dish with at least two stars lest I look like a real sop. The beads of sweat that subsequently moistened my delicate 20-year-old complexion conclusively betrayed my lack of experience with Southeast Asian cuisine, yet they successfully demonstrated a willingness to take risks and brave the consequences!

Fortunately none of my ordeal was in vain—I learned a short two days later that I had survived Microsoft's sacred initiation ceremony. I was in. Hooray! For the rest of my junior year, then, I sailed happily through my classes, feeling proud as ever with my induction into the software aristocracy.

To fuel my pride even further, I was assigned to the group working on an ambitious project called Laser. Laser was intended to be the ultimate solution to workgroup communications—the product that Microsoft would use to pinpoint, with deadly accuracy, a hugely successful competitor called Lotus Notes (eventually bought by IBM). And because this project was so new and so important to Microsoft's overall corporate strategy at the time, my assignment was—so I was told—a

very special and *very* important one.* Certainly it would provide ample opportunities for creative experimentation and *true* design work.

Through all of the following summer my performance was excellent. I once again threw myself entirely into the project, giving the best I had to offer for the greater good. I ably solved every problem given to me with precision and efficiency, qualities that drew frequent compliments from my team lead. And as I grew ever more confident about my future as a design engineer, I couldn't help but feel that I was taking my first steps toward someday reaching the pinnacle of Microsoft's technical ladder: Software Architect.

Yet I never once asked myself whether I should be climbing that ladder at all.

I knew, as described in Chapter One, that my place was Microsoft, so naturally I aspired to the most important technical role within the company. But was it what I really wanted, or what I really *needed?* It never occurred to me to ask the question. In my pursuit of career success I was doing what everyone else saw as "a good thing." And why not? The only alternative was life in the gutters of mediocrity, so to speak.

In our culture we recognize the ladder of material success. Imagine a drunken bum who one day decides to leave the bottle and earn a million dollars. Would we not applaud him? A life of

---

* Which was, I think, what interns were pretty much told about every assignment. "Microsoft's most important project," in fact, is always the one you are working on. Microsoft discovered long ago that internal rivalry is far better than envy. Laser, by the way, is the project I mentioned in Chapter Two that eventually crumbled into the much simpler form of Microsoft Mail, then gradually built up into Exchange and Outlook.

hard work and honest gain is infinitely preferable to mindless inebriation. Yet we also understand that a life of service or a life dedicated to high ideals is much more noble than one lived merely for personal profit. Seeking material gain in order to pull oneself off the streets is certainly a positive step…but to seek it when one has already found a higher calling? That's nothing short of a fall. Just imagine what the supermarket tabloids would say if someone like Mother Theresa, Mahatma Gandhi, or even a well-loved schoolteacher had gotten fed up with their work and opened a casino!

Life is about growth. It's about reaching upwards toward an ever-expanding vision of reality. But how this truth expresses itself for any one person is an entirely individual matter. Some need to learn how to work hard for themselves. Others need to learn how to transform such work into service. And some must then learn to serve not just people, but also higher principles or even Truth itself.

Now at the time I was an intern I already knew how to earn money for myself. What I needed, personally, was to move up a notch and learn how to open my heart to the needs of others.

Certainly my co-op job in Developer Support helped me in this way—I learned things as deeply as I could for the benefit of those who called in for help. And while the work wasn't considered glamorous it had given me the deep inner satisfaction of the noble warrior who willingly takes heavy burdens upon himself to spare others the pain.

But things changed while I was first working with the Windows team and during my internship. My star-studded role as a "real" software design engineer working on "important" projects blinded me to the point where I pretty much forgot about expansive ideals of any kind. In my pride as a Big Important Person, and in my narrow focus on the technical problems I had to solve, I lost sight of the fact that engineering is first

and foremost concerned with the needs of *people*. Those needs must be understood before an engineer can look for appropriate technological means to satisfy them. And by keeping the needs of others in mind, the work of engineering—and most other kinds of work, for that matter—can be a serviceful aid to one's own spiritual development. It's not the job, it's the attitude! What matters most is not what one does but the direction of one's energy in that activity.

At the end of my summer internship, then, I had this vague feeling that something was wrong. Yes, I had accomplished a great deal and had made a good name for myself as a programmer. But still…something just wasn't quite right about how I'd done my work.

"Bah!" I said to myself. "What nonsense—I know what I'm doing!" Thrusting aside these feelings of mine, I re-affirmed that I was on the right track: I asked my managers if I could continue working part-time on the Laser project while I completed my degree. This was, to my knowledge, an unprecedented request. Yet they were willing to give it a try. I only lived a short distance from Microsoft and could easily come in when I had the time. In fact, my class schedule that fall left me entirely free on Tuesdays and Thursdays.

This special arrangement was a tremendous boon for me. Besides the added income, I got to keep my office and access to all its state-of-the-art equipment. I also got to stay in touch with professional software engineers. And because I could just slip into being a full-time employee after graduation I got to avoid all that agonizing career-search business (which was absorbing a good part of my classmates' energies) as well as another interview!

My real motivation, however, was fear. I was utterly afraid to let go of my privileged position, lest it drift away and never return. I had to hold on at all costs…

For the next three months—the autumn quarter of 1989—my moderate course load allowed me several days of productive work at Microsoft each week. Everything seemed to be running along just fine. During winter quarter of early 1990, however, my load was dramatically increased by Dr. Kim's five-credit course on embedded computer system design, EE478.

EE478 was a legend in dread: whereas a typical five credit design class usually demanded about twenty hours a week for lecture, lab, and homework, EE478 demanded fifty. Combined with my other courses, it brought my peak class load to nearly ninety hours a week. Still, I found the course exhilarating: the project I chose fired my energy and enthusiasm and quickly became my first priority and my greatest passion.

Needless to say, my presence at Microsoft sharply declined down to the one day a week when I had no scheduled classes. The rest of the time Microsoft was condemned to roam amongst the shadows of my mind. As a result, I spent a fair amount of my one day in the office simply trying to remember what I had done the week before and catching up with everything the others had been doing. Even then I was constantly musing over my EE478 project—indeed, the very vibrations of Microsoft brought new inspirations for it! Worse yet—and this was a BIG mistake—I also started bringing my project to work. First I began coming into my office at night because the computers there were way better than the dinosaurs at school or the one I had a home. Then I started bringing my prototype with me so I could perform lengthy tests while trying to do my assigned tasks on Laser.

Well, you can probably guess what happened: I accomplished next to nothing as far as Microsoft was concerned. And what I did get done I didn't do well. Several times I carelessly

checked faulty code into our project just because I wanted to get it over with before leaving for another week. When the project couldn't be built, it delayed everyone else on the team; my development lead scolded me for my negligence and my office was the frequent home for our booby-prize "Bug of the Week" plaque that had this hideous purple rubber monster (named Slimer) attached to the front. I had to spend many precious nighttime hours coming in to correct my mistakes.

I really tried hard to do better. Yet more and more I found myself nearly incapable of giving Microsoft the energy and discipline that professional software engineering requires. For reasons I just couldn't comprehend, my mind refused to think about my Microsoft responsibilities, concentrating almost entirely on my EE478 project instead.

From a higher perspective, however, its clear to see what was truly happening. My being at Microsoft was wholly selfish: I was only there to protect my position as a design engineer and to stand aloof from other students who were deep into their job searches. My work in EE478, on the other hand, was completely different: I was working with a local school for the developmentally disabled to create a special device for their patients and staff.* In this my motives were completely pure. Indeed, it was the first time in my life that I'd really done anything resembling altruism.

---

* The device was a palm-sized timer module that plugged directly into a wall outlet and generated an audible signal at random intervals between certain preset points. The staff used this signal to check up on patients and evaluate the efficacy of treatments. A random interval was necessary: even the most mentally disabled patients would eventually learn the duration of a fixed interval to the point where they would behave one way for most of the time, then almost instantly settle down into a different behavior just before the monitor came to check up on them. (See www.mysticmicrosoft.com for photos of this device.)

Simply said, my self-serving attitude at Microsoft caused my heart to close and constrict. The opposite attitude toward my project in EE478 caused my heart to open and expand. Knowing which attitude would help me grow, the Universe conspired to flood my mind with those thoughts and inspirations that would lead me upward and resolutely shut out those that would keep me down. So whenever I tried to focus on my ego-motivated work at Microsoft, I expended nearly all my energy simply trying to resist this upward flow. Not much was left for technical tasks.

When EE478 finally ended and my spring quarter classes began, I still clung determinedly to my position at Microsoft, seeking to redeem myself with renewed effort. But the opposing tide was just too strong: though I had more time to give to Microsoft, my ability to concentrate continued to elude me throughout the term. Eventually my group managers had to seriously rethink my future with them. While I had proven myself a competent engineer the previous summer and was even slated to fill one of the available full-time positions on the project, it was obvious that something had shifted. On one level they could clearly see that I wasn't the right person for the project. On a more intuitive level they probably understood that the project—and the nature of the work itself—wasn't right for me either.

Thus at the beginning of the summer quarter, during which I was finishing up the last few classes I needed to graduate (the ones I'd missed during my co-op experience), they let me go. And with this my future as a big important software design engineer was over—not only with them but within Microsoft as a whole, for they couldn't rightly recommend me for the same kind of job elsewhere in the company. To my managers' regret as much as my own, I would just have to find some other position, if I was to even stay with Microsoft at all.

I was simply devastated—not so much by the decision itself but by what I considered to be the single greatest personal failure of my life. It was worse than the spelling test I flunked in second grade. It was worse than the linear algebra test I totally bombed in college. It was even worse than failing my first driver's license test in high school. For added to my failure was a shattered dream and the agony of being denied something so precious when I had come so close to having it for good.

Yet I wouldn't trade the experience for anything in the world. Having been stripped of self-importance I had to stop defining myself in terms of some label and its attendant rewards. No longer in a position to demand, I was given the chance to think in terms of what I could offer. And this one critical shift—this one fundamental change in my consciousness—opened up new opportunities that would carry me forward…on a path that led me once more into product support and yes—Lord have mercy!—even into marketing!

CHAPTER FOUR

Opportunity

"Its employees often work long, hard hours yet enjoy just about the most comfortable office environment around…call it a velvet sweatshop."

So ended (as memory serves) an article about Microsoft that appeared one day in a local Seattle paper. Whether the reporter's colorful analogy was meant to express admiration or not, we proudly took it as our hallmark. Within weeks someone had made up and distributed several hundred "Velvet Sweatshop" T-shirts.

But while this theme aptly portrayed what Microsoft *looked* like on the inside, it didn't do justice to the *energy*. For this, one might choose among many good adjectives: creative, dynamic, driven, and so on. But one word, in particular, stands out—a word that, to my recollection, was once and only once given its full, *proper* expression.

The unique historical event took place in the year 1990 at Microsoft's annual company meeting. Microsoft's revenues had just surpassed the highly symbolic $1 billion level. Everyone was thrilled. The late Frank Gaudette, Microsoft's well-loved Chief Financial Officer at the time, presented the news and, in his usual soft-spoken manner, repeatedly mentioned just how much he liked the sound of that word "billion."

Frank then turned the podium over to one who was, vocally speaking, his diametric opposite: Steve Ballmer, Microsoft's number two man behind Bill Gates.

It was Steve's usual form to come sprinting onstage and immediately turn the meeting into a boisterous pep rally.* We usually *heard* him, in fact, before seeing him. Today, however, he was clearly in a rare mood. He approached the podium with a deliberate, even dignified gait, and just stood there, calm and quiet, gazing about him with penetrating eyes until the thousands of geeks, nerds, and techno-weenies present were utterly silent.

Then with a quick inhalation he thrust his mouth at the microphone and thundered ***I N - T E N S E ! ! !*** In an instant we were all on our feet, simultaneously startled and inspired by Microsoft's principal cheerleader into the kind of celebratory tumult that the occasion demanded. I can't think of anyone who wasn't magnetized by Steve's enthusiasm! We all carried it back to work with us for months.

For most of the time I worked there, Microsoft's very halls radiated an intensity that I've rarely experienced anywhere else. You could just feel that it was *alive*, a place where new and wonderful things were being born every day. And you could see it in the people—the light in their eyes, the determination in their wills, and the joy in their hearts. They were the engines that powered the Microsoft Machine.

Of course, an engine that burns too hot will eventually seize up. Where people are concerned, the fires of enthusiasm and aspiration must be kept in balance: working for months with minimal sleep, little exercise, a poor diet, and literally no life to

---

* Video clips that exemplify this can be found on the Internet. Try searching for "Steve Ballmer"+dance.

speak of outside the office is a sure recipe for burnout. It goes without saying that many employees in the Velvet Sweatshop suffered this fate.

At the same time, many others went on for years and even decades without any signs of strain or fatigue. Why? Well, those who thrived within Microsoft's radioactive aura were the ones who realized that intensity—that is, being "in tension"—*must* be balanced by periods of relaxation. They intuitively understood that tension—a by-product of all striving—is not in and of itself a bad thing. *Prolonged* tension, on the other hand, has to be checked, for it invariably leads to over-exertion and exhaustion. Naturally, then, one must take an occasional break from one's professional responsibilities.*

But the solution here is not to just collapse on a couch and watch television for a week: passivity is only the negative counterpart to intense activity. Sure, it might feel good for a time, but in the end it only serves to deaden one's energy altogether rather than regulate it. *Regulation is the key*. As many of us at Microsoft discovered, the trick was to balance the intensity of one's work with *equally energetic fun*. Instead of fighting against the dynamic flow of energy that permeated our work hours, we willingly allowed that flow to energize our leisure activities as well. In doing so, we found that all of life became radiant with a certain zestful vitality.†

Indeed, many of those I knew at Microsoft could be truly classified as artists, even geniuses, in the field of creative re-

---

* Steve McConnell, a well-known author and software management consultant, reports that one of Microsoft's development groups asked that a washer and dryer be installed in their building so they wouldn't have to go home to do their laundry. Though it was clear that these people wanted to work, it was probably wise that this particular request remained unfulfilled.

† My own deepest experience of this is the subject of Chapter Ten.

laxation. The born-and-bred programmers among us actually relaxed with "recreational" programming—they dreamt up new computer games (like those that some sharp product manager turned into the profitable Windows Entertainment Pack) or hacked existing ones to make them more interesting. For example, there was a popular 3-D shoot-em-up adventure game called *Castle Wolfenstein.* Apparently unsatisfied with its original gothic setting, someone changed the graphics so that the walls, doors, and hallways looked just like Microsoft's.

Those who were slightly more athletic in temperament were quite inventive with games situated in Microsoft's real hallways. Swing Around the Wing, for example, was an after-hours golf tournament held inside one of the original two-story X-shaped buildings at our corporate headquarters. Given a pitching wedge and a putter, the goal was to send the ball in the fewest number of strokes around a complete circuit of both floors, including the stairwells and elevators. Another game was a variant of indoor lawn-bowling for which we used the rolls of masking tape and white-board pens that were readily available in the supply rooms.

Outside the corporate hallways, the truly physical sorts had a reputation for getting into the extremes of sport, including but not limited to bungee jumping, unaided rock climbing, helicopter acrobatics, and simulated aerial combat in private jets. Others who weren't quite so daring (or wantonly rich) stuck to the more traditional forms of volleyball, softball, and soccer, played on Microsoft's own full-size fields. Other perennial favorites included hacky-sack, Ultimate Frisbee, and team juggling.

There were also those who indulged in the more cultural pastimes of art, music, and theater, more often as participants, in fact, than spectators. Some joined the Microtones, our company choir, or were involved in the on-campus chapter of the

Toastmasters. Others formed an art committee that browsed galleries and selected works to grace the halls and reception areas of every Microsoft building. Others independently sang opera, acted in plays, or danced ballet.

Then there were those of us who, lacking stamina, taste, and the natural instincts of the recreational programmer, took to the more intellectual forms of play. These included complex strategy games played over the corporate computer network along with the late-night favorite of geeks and nerds everywhere: Dungeons & Dragons. This well-matured role-playing game was my personal choice. I first got into it during middle school, then picked it up again during my summer internship at Microsoft when I could play the game with (what I assumed were) more sophisticated adults.

Dungeons & Dragons, or D&D for short, is oriented around the typical milieu of fantasy fiction inspired by J.R.R. Tolkein: pseudo-medieval worlds full of dwarves, elves, wizards, warriors, evil lords, mysterious mystics, and an endless variety of creatures both benevolent and malign. Originally published a few decades ago as a small set of basic rulebooks, its steady popularity has since produced an Alexandrian library of new tomes, each adding another dimension of minutiae to the game. Far from being burdensome, however, the sheer intellectual complexity of it all was the very thing that attracted us!

Playing the game generally requires a group of four to eight people. One of them takes the role of Dungeon Master, the omnipotent creator and sustainer of some make-believe world. The creations of the DM, as he's called,* are limited only by his imagination and often go far beyond the basic town and dun-

* D&D was popular mostly with those of the male persuasion though not exclusively. For convenience I use the masculine pronoun here.

geon settings to include time-travel, space ships, and adventures in the subtle planes of the astral cosmos.

The others in the group, the "players," each create a character with varying proportions of physical strength, intelligence, wisdom, dexterity, constitution, and charisma. Like many other parts of the game, these attributes are determined randomly using dice. The resulting strengths and weaknesses determine the kind of role that your character can assume in his or her incarnation, be it a mighty warrior without a shred of common sense or a haughty intellectual necromancer who likes to talk but is a complete milksop where any real action is concerned. Or it might be like an immensely muscular tree-hugger type I once had who, though enormously wise in the ways of nature, was so uncivilized that his speech—and thus my own as the player—was restricted to fewer than fifty single-syllable words.

Everyone's characters are then set loose in the DM's world to make their fame and fortune or meet their doom. The players, not knowing what to expect, try to act according to the personality of their characters throughout each adventure. If your character survives long enough, he or she gains new skills, new powers, or in the case of my sylvan ranger, a larger vocabulary. From there one is ready for even more challenging quests.

The popularity of D&D (and other role-playing games that followed its lead, including online creations like World of Warcraft and Second Life) is due, for the most part, to its ability to accurately simulate the mechanics of real life without any of the dangers or constraints. In a world where so many people feel trapped by their bodies, their minds, their obligations, or various cultural expectations, games like D&D provide an attractive escape, allowing one to live out, to some extent, another life. In fact, players often create characters that reveal their own deepest yearnings. As my own hidden aspirations were essentially spiritual, for example, I gravitated toward

characters like monks, yogis, and others who relied primarily on their own inner strength. When (in my last year at Microsoft) I gave up playing the game altogether, it was because I finally decided to stop fantasizing about such roles and actually start living them!

A fascinating aspect of my experiences with Dungeons & Dragons was how often the players—myself as much as anyone—altogether missed the point. An essential ingredient in any adventure is the *unexpected*; what made the game truly fun was having to use all our skills and resources to overcome the challenges dropped on us by the DM. Like good movies, the best adventures were so intense that they pushed both players and characters to their absolute limits. Is that not where we experience the most growth? Indeed, it was the DM's job to create such situations. He was supposed to try his best, for example, to bring characters uncomfortably close to the brink of destruction before achieving a major victory, or contrive to strip a character down to his britches as a prelude to some fantastic reward. In these ways the DM hoped to make the game enjoyable—which was, ostensibly, the whole reason we played it in the first place! And all the DM sought in return was a little appreciation for his creative efforts.

However, we players often got it in our minds that the "world" was under *our* control: we constantly sought to alter apparently unfair circumstances by arguing with the DM. How we howled in protest when our characters lost some favored possession! How often we begged to go back in time and re-roll the dice when a pet character was killed in action! How often we threatened to quit the game entirely if the DM didn't make his world a little more favorable, even when all our (make-believe) problems were created by random chance or our own sheer stupidity!

The DM, of course, wasn't worth his salt if he gave into

such childish antics—*he* was in control, not the players, and unless he was downright cruel he almost always had some special blessing hidden beneath the surface. Admittedly, some DMs *were* cruel, forcing characters to fight their way through the nine layers of hell for a handful of pennies. But usually the DM simply wanted the players to have the courage and faith to look for the treasures he'd planted amidst the worst possible situations.

What usually happened instead was that the players, lacking such courage, just got mad at the DM—as a person, not the role—for his lack of fairness, giving him not love and gratitude but accusations and anger. Worse yet, players sometimes took things personally: friendships that were once shared outside the game itself became irreparably marred. And all of this over the roll of dice in a made-up fantasy land!

This is perhaps the clearest way that role-playing games like D&D successfully simulate our so-called "real world." Life certainly has its share of seemingly random tragedies. It has no lack of apparently unjust circumstances that we so desperately wish we could change or reverse. And we usually feel so powerless that there's little left to do but get angry with the whole mess and with the God who allowed it all to happen in the first place.

It's certainly how I felt in the summer of 1990. Where I had once prided myself on beating the odds, my coveted career as a Big Important Software Design Engineer were now decimated beyond any hope of recovery. Precious dreams had been taken away by forces that I didn't understand. I was stripped of an identity with nothing left to fill the void. And the timing of the whole incident seemed the worst possible—I was only a month away from graduation and all the while I had confidently expected to just slip into a full-time position at Microsoft. I hadn't even once thought about making back-up plans.

Yet in this desperate situation was a hidden grace: I simply didn't have the luxury of wishing or worrying. I had no time to deny the problem, nor did I care to even think of God, let alone argue with him. No, my only choice was to accept my circumstances *as they were* and to work *with* them, not against them. I had to immediately stop thinking about my problems and to start looking for solutions.

I once had a rather formidable warrior in a Dungeons & Dragons game who lost his favorite possession, an enchanted sword, without which he became quite vulnerable. But instead of backing off and contenting himself with a meeker role, he took it as an opportunity to renounce weapons altogether in favor of the martial arts. In doing so, he ultimately found that he became even more powerful—quicker on the attack, more nimble on the defense, and unburdened by what were once his "necessary" accoutrements.

Many of Microsoft's development groups have also taken a temporary setback and turned it into an unexpected victory. The most notable example I can remember is that of the Microsoft Foundation Classes (MFC), the set of programming tools mentioned in Chapter Two. The MFC team initially set out to match the features of a rival tool called the Object Windows Library offered by Borland International.* They accordingly based their core design on the same fundamental assumptions. After a year or so, when the team had produced a full working prototype of that design, further development was put on hold for what was called App Month. For one month everyone on the team used their new tool to create working applications, just like their future customers eventually would. But whereas they

---

* Now Borland Software Company who has since replaced the product with their Visual Component Library.

started App Month with unbridled optimism, they ended the month with the depressing realization that their creation was a bomb. Things that were supposed to be easy weren't. Things that were supposed to simplify tasks only added complexity. The list of flaws went on and on, and the whole year they'd spent on this dud seemed a total loss.

But the MFC team refused to see it that way. Instead of weeping over what they'd done wrong, they took it as an opportunity to question their basic assumptions. With the understanding they now had, they discovered entirely new ways to approach their goal. What they ultimately produced simply had no rival.

This is the consciousness of opportunity: instead of focusing on problems, focus on solutions. Every unexpected difficulty, every failure, and every seeming injustice are then merely stepping stones to eventual success. It might not be the success we thought we wanted, but always a success nonetheless. It's a simple yet profoundly powerful practice.

You see, we like to think that we understand the "big picture." We like to think we're in control. We like to think we know what's best. But who among us can truly make this claim? Who among us truly knows what's best even for ourselves, individually, let alone for the billions of other people and countless other forms of life on this planet? The universe simply resists such presumption. As a result, life seems to deal us an endless stream of "challenges" to which we normally respond with our own (often angry) resistance.

But these challenges are not challenges at all: they are invitations. They invite us to expand our vision. They invite us to set aside our cherished opinions about "the way things ought to be." They invite us to open ourselves to possibilities that we might never have imagined. The simple question is: do we accept the invitation? For just as the universe resists our

presumption, so also it *supports* our willingness to live in harmony with a greater purpose.

For me, total acceptance of my situation helped me let go of what I thought my career should look like, and once I let go, the right things started happening almost without effort. It wasn't long before I was able to look back on my "failure" and know that I just wouldn't have had it any other way. Indeed, as I continued to see the greater purpose of my experience unfold, I came to also feel that one sentiment that all invitations deserve: gratitude. For life's trials and tribulations are neither random fate nor divine punishment, but secret gifts from the Great Cosmic Dungeon Master, lovingly given to us for our joy. It is what God eternally wishes for each of us if we but have the courage to see it and the faith to embrace it.

CHAPTER FIVE

Leap of Faith

"Would you trust OS/2 to run a nuclear power plant?"

A man whom I will call Ken Johnson, a program manager in Microsoft's OS/2 product team, was interviewing me for a similar position in late July 1990. It was only one or two weeks after I'd been released from the Laser team (which had by this time been officially reduced to Slingshot) and I was graduating from college within a month. Thus I was exploring, with some anxiety no doubt, the available opportunities for a full-time technical position within Microsoft.

Bob Taniguchi, ever solicitous of my welfare, came to my rescue. Knowing that I wanted to stay at Microsoft if at all possible, he suggested two possibilities: one, a position in Developer Support where I worked with him as a co-op student; and two, a position as Program Manager in one of Microsoft's many product development teams.

This latter idea intrigued me. Program managers are deeply involved with creating new products. Though the work isn't as technical as programming, it's just as creative. This was especially true for an opening in the group working on a revolutionary new operating system called OS/2. Microsoft's strategy was for this technologically superior system to sup-

plant Windows and MS-DOS while maintaining compatibility with both. To that end, OS/2 was designed to look and feel like Windows, even down to the group of accessory programs to which my Calculator belonged. The OS/2 team was looking for a program manager to head up this exact aspect of the project and decide which new features to add.

On the strength of my experience with the Windows 3.0 accessories and Bob's recommendation (he was actually on the OS/2 team himself at the time), I was granted an interview. To my relief, the whole process was somewhat less intense than my previous experience: only three one-hour sessions this time, none of which were all that grueling. What's more, I seemed a relatively good fit for the work in question. So I felt increasing sure that I would get the job…and just think, I'd finally get to put that damn italic font into Clock!

That is, until Ken Johnson asked me about nuclear power plants. While I knew that OS/2 was already in the hands of real-world customers, I didn't yet trust the system enough to offer a qualified "yes." So I instead offered a tentative "no."

Oops. The sudden change in Ken's facial expression—and his revealing that two such facilities were already online with OS/2—instantly told me that I'd said the wrong thing. And not just wrong, mind you—I clearly did not have the necessary "faith," a character flaw tantamount to blasphemy. For at that time especially, Microsoft program managers really had to be *believers*. They had to believe not just in the company's overall goals but in their groups' product. And they had to believe, irrespective of any and all supporting evidence, that those particular projects were *the* most important things in technological history, important enough to inspire personal sacrifice—even martyrdom—when circumstances demanded it.

It hardly needs mention that I didn't get the job. But believe me, I was deeply grateful: in that group, martyrdom soon

became almost mandatory!

OS/2 was a so-called "cooperative" effort between Microsoft and IBM, both of whom were interested in retiring MS-DOS.* On this the two companies could agree. As for everything else? Well, let's just say that the partnership was a far cry from wedded bliss—the software development philosophies of the two just weren't compatible. Still, the two somehow managed to maintain the relationship long enough to produce offspring in the first version of OS/2. But that was all: during the development of version 2—and only weeks after my interview—the relationship ruptured. In what was a harsh divorce, IBM got full custody of the system and Microsoft's OS/2 team was left in shambles. So I was *very* grateful for having been rejected, especially as The Great Schism, as it was called, officially happened on the exact day, August 20th, 1990, that I would have started full-time work with that group!

I was even more grateful when I saw what happened next. Given this volatile relationship with IBM, Microsoft had secretly prepared for the inevitable. Even while Microsoft was working with IBM on OS/2 2.0, a small team hidden off in some lonely corner of Building 2 was quietly working on certain "new technologies" for what they ostensibly called OS/2 version 3. In reality the project had nothing to do with OS/2: it was rather the foundation for an entirely separate operating system designed to go head-to-head with OS/2 and win. When Microsoft and IBM broke up, then, Microsoft simply put its full energy behind this new system: everyone who had been working on OS/2 suddenly found themselves working on "Windows, New Technology," or, simply, Windows NT.

---

* Also known to IBM as PC-DOS. In the end it took more than another decade for Microsoft to finally shed the old MS-DOS code base.

Windows NT (now the core of current versions of Windows) quickly became the most intense project in Microsoft history, demanding the heart and soul of everyone involved.* People neglected their families, destroyed their marriages, and wrecked their health in sacrifice for the cause. I don't know what would have happened had I been involved myself. Without the necessary degree of unquestioning commitment to the project, I'm not sure I would have survived. All I can say is that I'm really, *really* glad that I wasn't given that job!

That left me with the opening in Developer Support. Since I'd worked there before and already knew many of the staff, my interview was mostly a time of renewing old friendships and catching up with the scene. There was a stout fellow named Dave Edson, for instance, who had been a co-op student at the same time as myself; we even shared an office for a few weeks. He had written a Windows version of the popular video game Tetris that was being included in Microsoft's first Windows Entertainment Pack. When I came to Dave's office for my half-hour interview he was just putting the finishing touches on a two-player mode. Under the pretext of helping him test this new feature, Dave proceeded to throttle me one game after another for a good twenty minutes. With time running short, Dave finally bothered to ask a few no-brainer questions to give the interview some semblance of formality; everyone knew I was perfect for the job.

A few days later I was offered a full-time position with a modest starting salary. Modest? Ha! Some would have said insulting. Given my experience I might have held out for twice as

---

* For the inside story of the inaugural Windows NT effort see *Showstopper: The Breakneck Race to Create Windows NT and the Next Generation at Microsoft* by G. Pascal Zachary.

much at other software companies and demanded stock options to boot. I probably could have found a design engineer position as well. My sense of responsibility, then, demanded that I investigate such options before making any decision about Microsoft. But in my heart I still knew that Microsoft was my home. Regardless of the role that they were able to offer me—and the salary—I just couldn't imagine being anywhere else. Perhaps I intuitively knew that most of those other companies would also be gone in a year or two! Whatever the case, I just wasn't interested in shopping around for another employer any more than I thought to shop around for another spouse. It wasn't necessary: I had been given what I needed, and it was now up to me to make the most of it.

I thus made what seemed an entirely impractical decision: without hesitation, regret, or concern for the low-rung nature of the job and its meager salary, I accepted Microsoft's offer.*

Foolish? Perhaps. Irresponsible? Definitely. But based on what followed in its wake, it was one of the best decisions I ever made. For one thing, the pain of my failure to become a Big Important Software Engineer simply vanished. In its place blossomed that same joy in helping others that I'd felt during my first months at Microsoft, a joy that had been absent for nearly two years. Every week I seemed to learn more than I had in my whole fourteen-month stint as a programmer. And my self-confidence, so recently shattered, both recovered and was growing stronger by the day.

---

* At the time, Microsoft typically started new hires at a salary level lower than the industry average, sometimes substantially so. However, unlike most companies who were considered generous to offer a raise of 3% every one or two years, Microsoft offered up to 6% twice a year, exceeding that rate when extraordinary circumstances demanded it. On top of that, Microsoft had its stock option program, legendary for creating millionaires by the boatload (see Chapter 15). One eventually had little reason to complain.

This was also a time when I was blessed with a number of new and lasting friendships. The one that holds a special place in my heart was with another new-hire named Charlie Kindel. Charlie and I started within a couple weeks of each other and shared an office for the first few months of our parallel careers before a departmental reorganization put us in different teams. Later on, however, we worked together again, then were separated, reunited, separated, and reunited at least three more times. In fact, Charlie and I worked together in every group of which I was part until I left Microsoft six years later.

In my work itself I made rapid progress: within only two or three months I was considered one of the best support engineers in our whole department. I seemed to have a certain knack for our particular *métier*—not only was I able to understand a wide range of technical details and apply them to specific problems, but I was especially adept at the more difficult task of clearly communicating those solutions. On top of this I seemed to have an innate ability to quickly generalize a very particular solution and apply it to an much broader range of questions. Written up as short articles, these generalizations were especially valuable as part of our electronic Knowledge Base, one of the central resources in our support work.*

Best of all, I found a special joy in working with the people who called in for assistance: I solved problems to help people, not just to solve problems. Helping others succeed gave me a satisfaction that no strictly technical work really could. It was so satisfying, in fact, that it became the cornerstone of all my remaining years at Microsoft.

---

* This database is publicly available today as the Microsoft Developer Network—http://msdn.microsoft.com. A number of my articles from these days still appear there, as well as other articles and papers I produced throughout my career.

All this stands in marked contrast to what I probably would have experienced as part of the Windows NT project. I suspect that the role for which I'd interviewed would have been pulled out from under me—God only knows what my new assignment would've been! But I can tell you this for certain: I would have very likely been quite resistant to the whole upheaval, especially when it became clear that Windows NT was ready to consume every ounce of energy that one was willing to give, and then some. As a result, I would've held part of myself back and kept my eyes open for an escape. This, in turn, would have limited my effectiveness even as the project made ever-increasing demands on my life.

Like so many others, I would only have been motivated by the hope of some future reward that makes present suffering bearable—that same kind of unquestioning "faith" that we commonly hear about, especially in religion. Such faith, however, is based solely on the fear of losing the reward. While it might inspire one to heroic (or stupid) degrees of self-sacrifice, there is little love or joy to be had along the way. For myself, I can't go on very long without that love and joy. In the end, I probably would have become one of the many unfortunate casualties left in the wake of Windows NT.

In Developer Support—with my meager salary, no stock options, and an unspeakably minor role in Microsoft's overall success—there were no such promised rewards. I truly had nothing to lose, nothing to fear, and no need to look for an escape. The only fulfillment to be had was in the present, and the only motivation was simply the love I felt for the work itself. With nothing to hold back, my effectiveness was extraordinary.

In this came another opportunity, one that would eventually play a very important role in my inner transformation. Unlike most positions in product development, which had this

annoying habit of making martyrs, my sort of job in Developer Support demanded little more than the usual forty hours a week. This left me with far more free time to invest in other pursuits than most software engineers and program managers could hope for.

And what were those pursuits? Among various interests and hobbies (such as creating or playing in some Dungeons & Dragons campaigns) I essentially spent much of that time—oddly enough—preparing myself to one day *leave* Microsoft.

You see, though I was totally committed to Microsoft and knew that it was where I belonged at the moment, I also knew deep within myself that I wouldn't be there my whole life. I had only entered the computer field, you might recall, because it offered the best opportunities—not because it was my true passion. What really interested me were things like astronomy, history, cosmology, music, psychology, photography, and even certain elements of spirituality. To these interests I someday hoped to devote more, indeed, *all,* of my energy. But I didn't want to become a starving artist who was forced to accept any old commission out of desperation, nor did I want to become a starving scientist who was similarly forced to work for ignoble ends. No, whenever I was ready to explore a new direction I wanted to do so with a certain degree of financial independence. My primary purpose in being at Microsoft, then, besides having fun with computers, was to save up enough money to make it all possible.

Until then, I could read. With so much time on my hands, including the hour or so each day that I commuted by bus, I began diving into all kinds of books. It was only now, in fact, that I really began to read seriously. I hadn't been much of a reader in my youth, and during college I rarely got to crack a book that wasn't the pet favorite of some professor with a penchant for the abstruse. Since high school, however, I had

collected a number of attractive titles "to be read later." It was now "later."

I thus began a literary adventure that would pass through several hundred books over the next five years. What exhilaration I found in the free air of new ideas! Without the pressure of homework or the looming specter of an exam, I was finally able to read what I wanted at whatever pace I wanted and actually think about it in ways that were personally rather than academically meaningful. Ah! Such joy, such joy!

My journey began with *Mathematics and the Imagination* by Edward Kasner and James R. Newman, which deeply inspired me with its far-reaching examination of concepts like time, dimension, and infinity. Then I launched into a nine-month expedition through the fascinating tapestry of humanity, guided by H. G. Wells' monumental opus, *The Outline of History*.

Written to be concise but complete, *The Outline of History* offers many profound insights into the development of government, warfare, science, and religion, to name a few. With Wells' rather universal approach to the latter, especially, I began to understand not only religion's outer differences (over which most of the wars in the book were fought!), but also its inner unity. Here I began to see that certain spiritual principles exist independently of particular religious forms. Here I began to glimpse a Truth that transcends both religious dogmas and the institutions built to promogulate them. And here, for perhaps the first time in my life, I discovered the freedom to ask questions and explore lines of thought that were all but forbidden in my parochial upbringing. In short, *The Outline of History* awakened my spiritual search—a search for Truth. Almost every paragraph in the book sparked new questions in my mind that demanded answers. In fact, just to work through those questions I once considered writing a commentary on the

book itself—a labor of decades! Somewhere in all this history, I thought, there must be some kind of *true wisdom*, some guidance as to how one should not just think, but how one should really *live*.

But it was not the time to attempt a synthesis—it was the time to just discover ideas and collect my thoughts about them, most of which I scribbled in the margins of each and every book I read. Someday I'd be ready to pull all of them back together into some kind of coherent picture. Then, I felt, Truth would reveal itself.

In the meantime, my creative and analytical energies continued to pour into my daytime work at Microsoft. Month after month my productivity soared above expectations. In one three month period, for example, I wrote nearly a hundred new articles for our Knowledge Base, fully twenty times my group's average. I also typically answered half again as more customer questions than the average. I even began working on articles for our company's popular technical magazine, Microsoft Systems Journal (now MSDN Magazine).

Then, only eight months after I'd started and despite my being the youngest employee in the whole department, I had developed such an expertise that I was one of four engineers chosen to start an elite Premier Support Group. We were entrusted with the task of personally guiding the development of new Windows applications by the other leading software companies of the time—including Microsoft's direct competitors. My most important client was WordPerfect Corporation who was creating the first Windows version of their highly popular word processing program (now owned by Corel Corporation). My dear friend Charlie Kindel, who had also been chosen for the Premier team, was similarly assigned to Lotus Corporation and their legendary spreadsheet 1-2-3 (now part of IBM's Lotus SmartSuite).

Odd as it was to help Microsoft's rivals build products that would challenge our own, it was our deep joy to help other programmers in this way. Ironic, too, was this particular destiny for me. After having so deeply desired some kind of "important" software engineering role within Microsoft, and after being forced to let it go, I suddenly found it fulfilled: the competitors' applications we were helping to create were actually vital to the future success of Windows (which needed a healthy third-party software market) and thus to Microsoft as a whole. One could even argue that our individual work was *more* valuable to Microsoft than that of most individual product development engineers. Indeed, our upper managers seemed to agree: within a year my scanty starting salary was raised three times (as was Charlie's), once by an astounding 25%; everyone in the Premier team also received a generous grant of stock options equal to what was given out within most product teams.

Thus only months after my great "failure" and what looked like an utterly foolish decision to stay with Microsoft, I found myself with far more than I would have ever expected or been able to find elsewhere. The key was the simple act of complete self-offering—a leap of faith, as it were.

A friend of mine, after he'd completed a major undertaking against countless odds, was asked how he did it. "Faith in God," he replied.

"Well, sure," the interviewer retorted, "but wasn't there a need to be, you know, *practical*?"

"Listen," my friend said, "I've found that faith *is* the most practical thing of all!"

The greatest success in any endeavor comes when we can focus all our energies in one direction. As much as we deify the gods of Reason, Logic, and Due Consideration—as befits our culture's scientific bias—and as well as they seem to work for us, we're always left with some degree of uncertainty. Did we

really make the best choice? Was there something we missed? Was there another path we might have explored more fully? These nagging doubts simply drain us of energy that would better be applied to whatever goals we're trying to achieve.

The positive expression of faith—that which is based on love, not fear—overcomes such doubts. True faith brings one's attention to the here and now, never regretting the past, never merely hoping for an uncertain (or unprovable) future, and never wasting any energy looking around for alternate routes. True faith is the conviction that fulfillment will be found by going *through* whatever Life has set before us, even if we cannot see where the path is going. While we often think it necessary to impress God by our suffering and self-sacrifice in the name of some belief, spiritual growth has nothing to do with convincing God of anything. His blessings and guidance are always there. He wants our happiness! He wants our fulfillment! We just can't be attached to the *form* of that fulfillment.

Be open, then, to new possibilities; let Life—let God—lead the way, no matter how strange or silly it seems. For with that simple faith we find ourselves guided, step-by-step, over every dark ocean of uncertainty to the shores of new and wondrous worlds.

CHAPTER SIX

Esprit de Corp

"Excuse me…do you know you're moving today to Building Four?" A burly gorilla of a man stood in the doorway of my office. He sported the uniform of Graebel Van Lines, Microsoft's more-or-less resident moving contractor.

I had no idea I was being moved, so I called our group assistant. "Oh! Um, yes…" she replied, a little embarrassed, "I forgot to tell you about that. Sorry!" Well, there wasn't much left for me to do but pack up everything as quickly as possible and take the rest of the afternoon off.

Experiences like this one, which took place somewhere in 1992 if I remember correctly, were not at all uncommon. Office moves—both expected and unexpected—were just a fact of life. We called it The Microsoft Shuffle.

Microsoft began its corporate existence in 1975 with two employees—Bill Gates and Paul Allen. Microsoft was initially centered in the Sundowner Hotel in Albuquerque, New Mexico, close to a company called MITS for whom Bill and Paul were writing a version of the BASIC programming language. Within three years, Microsoft had grown to fifteen employees and now included a brilliant software architect by the name of Gordon Letwin.

Around 1980 or so, with a few dozen more employees in the ranks, Microsoft moved its operations to the Seattle area; Bill's original stomping ground. They settled nicely into an office building situated at the intersection of Interstate 405 and State Highway 520, in the city of Bellevue. It was here that both Microsoft Word and Microsoft Windows were born.*

As the company continued to expand, Microsoft soon filled the entire building and Gordon Letwin was becoming, so the story goes, more and more nervous. He was apparently concerned that a large earthquake—the "big one" area seismologists continue to predict—would level the five-story structure and produce, in his words, "techie pancakes."

So when Microsoft began to build its Corporate Campus in the nearby town of Redmond, its expansion went primarily horizontal: the six original X-shaped buildings, each designed to maximize the number of window offices, were only two stories tall and built like fortresses. The massive pillars in the single-story underground parking garages gave unmistakable testimony to this fact. The only pancakes to be found at Microsoft were those in the cafeterias!

Of course, within a few more years Microsoft was again bursting at the seams. When I joined the company in 1988, many employees were doubled up in what were designed to be

---

* Up through Windows version 3.11 there was this special tool called Heapwalker (*a.k.a.* Luke Heapwalker) that allowed programmers to look at the way Windows organized the computer's memory (called the "heap"). Heapwalker always showed this one memory segment called Burgermaster. For years people wondered what this really meant. Finally the software engineer responsible for it fessed up: the contents of the segment itself were so uninteresting that he'd had a hard time coming up with a name for it. For lack of anything better, he christened it after the drive-in hamburger joint that he could see out his window. When the story finally broke, a photograph of the restaurant's distinctive sign, which exists to this day, appeared on the front cover of Microsoft Systems Journal.

comfortable one-person offices. Thus construction began on buildings 8, 9, and 10. Microsoft also purchased a number of other adjacent lots that were already home to what became buildings 11 through 15.* In addition, a number of the more self-contained divisions were moved off campus entirely, including the whole of Product Support. But still, no Microsoft building was more than two stories tall.†

Then Gordon Letwin retired and Microsoft's planners were finally free, so it seems, to go vertical. Starting with Building 16, three stories was the minimum. By Building 28 it had crept upwards to four. Last time I visited Microsoft the ceiling had obviously (and literally) been pushed to five. And somewhere back in 1991 Product Support took over a number of floors in a downtown Bellevue high-rise. Today, I believe, Microsoft has taken over the entire building and part of the one next door.

As you can imagine, Microsoft's endless expansion meant that we were always moving, moving, and moving again—into new buildings, into old buildings, within the same buildings… anywhere and everywhere. And by "we" I mean yes—entire departments and even entire divisions! The capable folks at Graebel Van Lines, in fact—who were and still remain literally a fixture on campus—developed the ability to shuffle well over a thousand offices in a single weekend. Their moving trucks are a constant hazard in Microsoft's serpentine parking lots (*a.k.a.* The Microsoft Speedway), and piles of their moving boxes and rolls of packing tape ubiquitously adorn every floor of every Microsoft building. They just never disappear.

---

* Since leveled and replaced with Microsoft's Conference Center.

† You might have noticed an omission here. It was a standard joke to send new hires to an "important meeting" in Building 7, which was never built because of setback restrictions.

During my eight-and-a-half years at Microsoft I occupied no fewer than thirty-three separate offices, both on and off the main campus. No joke! That's an average of one move every thirteen weeks. The longest I ever stayed in one place was sixteen months; the shortest, two days. In that instance I had just finished unpacking when I was told I'd been moved into the wrong office!

Most moves, of course, were flawless: we knew well ahead of time when and where we were moving. On occasion, though, as I related at the beginning of this chapter, you didn't find out about a move until the Graebel Gorillas showed up to haul everything away. It's times like this when packing is most effectively accomplished with a shovel.

I don't really remember anyone ever being upset by a move. For one thing, you always got a Friday afternoon off without having to dip into your vacation hours; when a really big move happened you got all of Friday plus half of Thursday. Moving offices also offered a much-needed opportunity to sort through piles of accumulated papers, product specs, outdated software, and outright trash. For some, the first day after a move was the only time they actually saw the surface of their desks.

More importantly, I think all of us understood that The Microsoft Shuffle was an important part of a deeper and more fundamental fluidity. In the fast-changing world of personal computer software, Microsoft had to be able to stop on a dime (or at least a few million dollars) and launch off in some new direction. Throughout its history, Microsoft's complete willingness to rearrange its internal structures has been vital to its continued prosperity.

And Lord, were we willing! Reorganizations had to count as one of Microsoft's favorite corporate pastimes. Sometimes they happened so often that we referred to the latest instance as the *"reorg du jour."* But we were glad to have them—they kept

Microsoft from becoming too fixed in its ways. (Isn't it fitting that many old companies are referred to as "firms"?) Microsoft's directors clearly understood that corporate structures and a well-tuned management chain are only of secondary importance to creativity, innovation, and service. You could, in fact, work for years at Microsoft without ever being aware of your position in the corporate hierarchy—or, as we called it, your Distance from Bill. Even when you *were* aware, it had little, if anything, to do with your actual responsibilities.*

Microsoft discovered long ago that the whole purpose of organizational structure, as with any structure in one's personal life, is merely to facilitate the flow of energy and creativity toward one's highest goals. If structures were found to block this flow of energy they got thrown out and replaced with something else. Those that did work often got a makeover just to see if they could be made to work even better. It even happened to development teams that were only months away from shipping a product! Truly, no structure was sacred.

The same was true about *how* we worked: methodologies seemed to change with the seasons as we constantly tried new and hopefully better ways to fulfill our various responsibilities. Development teams, for example, were always trying out new programming disciplines (or non-disciplines as the case may be), not to mention new ways to manage increasing complexity. Marketing groups, for their part, tried out sometimes vastly different presentation styles. They even once launched an operating system, Windows for Workgroups version 3.11, with a

* It was our common amusement to watch our Distance from Bill change from week to week according to the latest shuffles in upper management. During one three-month period, for instance, I moved from being six spots away, up to only four, down to five, then down to seven while nothing about my work or my immediate management was affected in the slightest!

full-fledged Broadway musical comedy. Note the word *once*.

In Developer Support we were always looking for ways to answer questions more accurately and more efficiently. And just as often we up-ended how those questions came to us. At first, programmers could call us for free to ask questions. Then we eliminated the direct phone lines and required that every question be sent through a fee-based electronic service. Then we opened up the phone lines again for companies that signed large support contracts. Then we had a complete restructuring where the whole of Developer Support went back to the phones with yet another kind of fee structure.

For a long time I never understood why Product Support's upper managers couldn't just make up their minds and figure out the best way to do things. Just when our work seemed to be forming a comfortable groove, they'd up and change it again! Finally I came to see that there was no "best" way that could be firmly set in concrete: the volatile nature of the software market simply demanded that we be as fluid as the rest of the company. The directors of Product Support were thus always shuffling us around in anticipation of the next storm.

In fact, keeping the energy fresh and dynamic was perhaps the real job of Microsoft's middle and upper managers. On some level they each understood an important principle: when energy is flowing in the right way, the necessary structures will naturally follow. I don't recall a single instance in my whole career when anyone—from Bill on down—talked about the importance of *how* things were organized; instead, they constantly encouraged us to work intelligently and energetically, with an incessant focus on our core mission: improving people's lives through technology.

This strong flow of positive energy toward a single, high purpose has been, to my mind, the most important factor in Microsoft's stunning success. As long as Microsoft can keep the

energy flowing upwards, it will continue to succeed—regardless of whatever competitors or lawsuits come forth to challenge it.

To understand why a strong flow of energy leads to the kind of success Microsoft has enjoyed, whether applied personally or professionally, consider the fundamental principle of electromagnetism: an electrical current flowing in a wire generates a magnetic field around that wire in direct proportion to the strength of the current. That magnetism has the power to draw to itself those materials that resonate with its field and to repel those that are its polar opposites. What the field attracts and what is repels is entirely a function of the intensity and the direction of the electrical current.

Similarly, the flow of energy within an individual also generates a kind of magnetism. But instead of attracting lumps of iron and such, that magnetism attracts those thoughts and inspirations that resonate with the direction of the flow, similarly repelling their opposites. Think about it for a moment—isn't it true that when you feel "down in the dumps" your inner energy, in the spine especially, is literally flowing downward? Isn't it true that such a state of mind attracts almost nothing but negative thoughts? Now consider the opposite feeling: when you feel "up" or "high" or "positively buoyant"—when energy is flowing up the spine—it's hard to think anything but the most joyful thoughts! You can even see the effect of that flow in the body: a depressed person hangs his head, hunches over, and looks at the ground; a happy, uplifted person holds his head high and looks to the very vaults of heaven.

The interesting thing about this kind of magnetism is that it depends on the *kind* of energy expressed, as well as its direction. Its operative field is consciousness. Actions and attitudes that seek prosperity will *attract* prosperity. Looking upon everyone as your friend will *attract* friendship. And a focused search for solutions will simply *attract* solutions. One time in

my later career, for example, I was working on a conference presentation in which I had to talk about certain aspects of "object-oriented software design." To finish the presentation I only needed to describe the steps of a certain design process, but all I really knew of that process was the mere fact of its existence. Under normal circumstances, then, I should have done a little research into the matter by skimming a few books in the corporate library and finding someone within Microsoft who was in the know. Unfortunately, with only an hour or two before the overnight mail went out (tell me *you* never procrastinate!) I had to try another approach. I went for a short walk on the forest trails behind Microsoft and did my best to attune myself mentally to the process and to the people who understood it. Within ten or fifteen minutes I had what I needed: all the steps became perfectly clear in my mind. I returned to my office, finished up my presentation, and sent it off.

Rather cheeky of me, wasn't it? I had no logical reason to know what I was talking about. At the same time, I *did* know—not by virtue of experience or study, but by attunement of consciousness. I felt it intuitively. And this feeling was justified when I gave the talk a month later: people afterwards told me that those steps were both correct and that I had articulated them *very* clearly.

In more recent years, I've found this same approach very helpful in a wide variety of other activities, from wiring circuit breaker panels and cooking to composing music and working with school children. A great deal can be accomplished with energy, intuition, and sensitive attunement to the task. Time and time again I've been positively amazed by the results.

Even more amazing, perhaps, is that this same principle also holds true for an organization, that is, groups of people. Great things are possible when individuals come together and direct their energies toward a common purpose.

Case in point: not long after I started my full-time job in Developer Support, The Microsoft Shuffle moved our whole group to a new office building about a mile away. Up to that time, all groups within Microsoft's Product Support Division were housed together in a cluster of three buildings called Lincoln Plaza. Our move to Ridgewood F, as it was called, only involved those of us who supported programmers. In this new building, well away from the concerns of end-user support both physically and psychically, we were able to focus more clearly on the unique issues of software development. This step alone immediately drew to us a deeper understanding of our work and new inspirations about serving the needs of programmers.

Of course, the move couldn't happen without the requisite reorganization! My department, which dealt specifically with Windows programming, originally had three separate teams corresponding to the three main architectural divisions of the operating system. Many problems spanned those boundaries, however, meaning that they could only be partially answered within any one team before being passed to another. It was a great source of inefficiency and delays: even simple problems could take days to answer. The more complex ones languished within our system for weeks.

Our teams were thus re-designed to operate independently from one another: each contained the necessary expertise to answer just about any question. As a result, we provided far better and timelier support and also developed, as individuals, a deeper understanding of the whole. Before, whenever I had encountered a question I couldn't answer, I simply punted it (according to standard procedure) to the team that could. It got questions answered, but I never saw the solutions and thus developed little knowledge of those other areas. Under our new arrangement, I took these problems to another teammate who could explain the solution to me. It was then my responsibility

to work through that solution myself and write up the answer for the customer.

This way our individual teams (and individual engineers) each grew stronger in themselves. Our new efficiency also gave us time and energy to develop more extensive specialties—user interface, multimedia, and so forth—as well as unique personalities! My group specialized in user interface and became famous for its toys and widgets—you had to be careful walking through our area lest you make a friendly acquaintance with a stuffed animal gone airborne or with one of the Nerf Rockets that were occasionally launched from my desk. We also sponsored office sports like Nerf Baseball and carpet bowling (using whiteboard pens and the omnipresent rolls of packing tape) in the still empty parts of our new building. The team that specialized in multimedia, on the other hand, became known as The Jungle: besides spending a good portion of their budget on fancy new audio and video hardware, they invested generous funds for large, tropical office plants. Accordingly, they set up their computers to generate a wide variety of bird song amidst a soft Amazon rainfall.

Anyway, our new arrangement worked out magnificently. Spirits were high and productivity better than ever—better than anyone expected, in fact. After our reorg, you see, we brought in a number of new engineers for each team. Typically it took four to five months before such new hires could handle a normal workload. But in just about every case they quickly developed an expertise that was totally out of proportion with the time they spent studying. With only moderate effort on their part, and in half the expected time or less, they somehow knew the answers to all kinds of questions, even ones that none of us had ever seen before. It was really quite amazing.

I don't think any of us noticed how extraordinary this phenomenon truly was: we were all so immersed in the magnetism

of our group that it just seemed perfectly natural. Then again, it *was* perfectly natural! When anyone entered into the magnetism of our highly energetic atmosphere, they could not help but be uplifted. And as our energies were directed toward solving every question that came to us, even the newest engineers found their minds literally brimming with solutions. We simply attracted that consciousness.

This kind of uplifting magnetism—the *group spirit*—is the same that one feels when entering a special meeting, a championship sporting event, or a sacred ceremony. It's real and it's tangible. With openness and receptivity, it only takes a little effort to tune in and absorb the inspiration that permeates the very air. It's what I've always felt at Microsoft and it's what we all felt in Developer Support: the subtle but powerful blessings for those who join together in harmony for a shared ideal.*

One day, in fact, we witnessed the effect of this unifying spirit on a group of people who were as likely as anyone to be wholly put off by Microsoft's typically unorthodox ways.

When we moved to Ridgewood F the latest method (*du jour*) of providing developer support was to accept all questions electronically. Every afternoon at 2 o'clock, all our teams gathered together for "triage" where we divvied up the new requests.

With no conference room large enough for us (we numbered about three dozen), our triage took place on the open floor in an

---

* The influence of environment is equally present, it must be noted, when a flow of energy is directed toward negative ends. In a negative environment, great vigilance and will power are necessary to resist being pulled down. You've probably experienced how very easy it is to become cynical when hanging around cynical people or even one exceptionally cynical individual. The same holds true for all other negative influences like anger, restlessness, materialism, lust, selfishness, etc. The influence of media is also very important to consider, *especially* that of music because it so easily bypasses the rational mind to affect you at the very heart of your being.

empty part of our new building. Some of us brought our toys and stuffies; in the distance we could hear the soothing sounds of the rainforest. Gathered in a rough circle, some of us would sit cross-legged, others would lean against the walls, and a few would stretch out in comfortable but somewhat undignified positions. Here, also, the utterly casual nature of our attire revealed its full glory: though some were dressed decently, others sported badly torn blue jeans, disintegrating running shorts, or grody old T-shirts that, theoretically at least, had seen better days.

It was during one such triage that a group of very corporate types—you know, real "suits"—came to tour our facility. Their company was considering a fairly extensive support contract and wanted to see us in action. Had we been told of their visit we might have dressed up and behaved ourselves better. As it was, they got to see us sprawled out every which way, papers all over the place, looking patently disgraceful. But what could we do? We had no choice but to continue our meeting as usual, working through all the new questions with every skill we had.

We later learned that our visitors had signed the contract because they felt our spirit and the depth of our commitment to service.

And they even described our pell-mell triage as "the most *professional* thing we saw."

CHAPTER SEVEN

A Bigger Pot

"I'd love to have ten of you. Would you like to be cloned?"

Dan Quigley, manager of the Premier Developer Support Team, was sitting with me for my August 1991 performance review. As I have already described, helping other programmers gave me a deep inner satisfaction and a special joy that inspired me to an extraordinary level of productivity. Dan thoroughly agreed and said so on my review. The subsequent raises and bonuses I received were nothing short of extraordinary themselves.

At the same time, I wasn't particularly happy with the outward circumstances of my job, so much so that my response to Dan's idea of corporeal duplication was that "ten of me would *not* work here." Though my comment saddened Dan a bit, he was not insensitive to my struggles and did his best to brighten my spirits.

Both of us were trying to understand what had changed. A short twelve months earlier I was a bright-eyed, enthusiastic young new hire who rapidly became one of the best engineers in our entire department. But now I was becoming more and more cynical by the week—not with the ideal of helping other programmers, mind you, but rather with the particular organ-

izational construct (Developer Support) through which I was trying to manifest that ideal. I loved my work but came to despise my workplace.

The situation here was not unlike many a Dilbert cartoon. Friends who have not experienced the corporate life sometimes ask whether the stuff they see in that comic strip has any basis in reality. "Well," I reply with a chuckle, "just about all of it's true—only slightly exaggerated!" Indeed, most Dilbert strips are inspired by real-life stories about the outrageous quirks of the modern corporate scene, especially those involving the all-too-common failure of employees and managers to see eye-to-eye on just about anything.

Perhaps Nature intended it this way; if nothing else, it at least adds to the richness and diversity of life. Managers often have tens if not hundreds of employees under their wings, not to mention endless worries about budgets and such, making it practically impractical to really understand the perspective of any individual worker. Workers, for their part, are notoriously small-minded and/or selfish, concerned mostly with themselves and their particular jobs in the here-and-now. So when management decides to invest in brand-new office furniture, for example, to "make everyone more comfortable," those who are already content say, "Well, if you really want to make me more comfortable, how about giving me a pay raise instead?" And so it goes on.

Being all of 22 years old at the time and still full of youthful immaturity, I was not only unable to relate to what Developer Support's upper managers had to contend with, but was pretty much unwilling to even *try*. In my mind, what I wanted and what I felt was right were all that mattered: I wasn't exactly open to other points of view.

I had started my Developer Support job with the correct understanding that customer service in our line of work meant

solving the multifarious problems that those customers sent to us every day. Therefore we answered questions over the phone, answered service requests (as we called those questions that came in electronically), and disseminated our growing expertise through Knowledge Base articles. In this latter way, especially, we hoped to create a resource where programmers could find many answers on their own. So far, so good.

In my mind, however, this was only the beginning. I tried to write articles that would guide people through complex tasks from start to finish rather than addressing only specific troubles. In my sample programs, too, I did my best to show how something was done and to offer self-contained code that other programmers could just drop into their own projects as-is. In time, I thought Developer Support might even make its own business out of small, reusable software components that addressed our customers' most common difficulties. I figured that if Developer Support could take on the task of writing and testing such components, it would save other programmers the trouble and, in turn, greatly reduce the overall volume of support questions. In time, we might make it so simple to write Windows applications that the need for developer support as we understood it could be wholly eliminated. And by so making our present jobs obsolete, we'd have the freedom to pursue even greater things.

These thoughts infused my assigned duties with a passionate sense of purpose and direction. For such goals I was willing to make some of personal sacrifices that I would not have made for other projects (like Windows NT, see Chapter Five). Still, I recognized that my ideas were somewhat revolutionary and best kept to myself: it wasn't my responsibility, after all, to set a direction for our department! I could only nurture the hope that those in charge would, in time, take notice of what I was doing and become interested themselves.

Unfortunately, I was also rather attached to this particular outcome; so attached, in fact, that I was utterly blinded to the special recognition I *did* receive for my work. Attachment is, indeed, blinding, and in my case it also caused me to place the worst possible construction on a variety of wholly benign circumstances. As a consequence, I felt increasingly thwarted in my aspirations and thus increasingly angry.

It started small, literally, with a somewhat minor award that I didn't get yet felt I deserved. Around the end of 1990 upper management began to recognize individual excellence at our monthly departmental meetings with these precious little six-inch faux-marble pyramids embossed with an appropriate motivational maxim. For the first few months I cheerfully applauded the recipients of the award. It was obvious that they had gone above and beyond the call of duty. Naturally, then, I felt it wouldn't be long before I was also honored—I handled far above the average number of calls and service requests each day. I wrote more Knowledge Base articles than most everyone else combined. And Microsoft Systems Journal had recently accepted a more extensive article of mine for publication. This alone was a significant achievement for *any* engineer within Microsoft, let alone a *support* engineer.

Yet month after month I was passed over, even after the list of other notable individuals was wholly exhausted. The awards were now going to people whose performance, from even the most unbiased perspective of raw statistics, couldn't shake a stick at my own! I just couldn't understand it. I seemed to be the only one whose special efforts went unnoticed, and it hurt.

Had I bothered to ask anyone about this, I would have easily discovered my being selected to help found the elite Premier Support Group was a much greater reward than some plastic pyramid! But I never did ask. Cursed, you might say, with an annoying tendency to stew over baseless assumptions about the

intentions of my superiors, I have, on occasion, allowed myself to become frustrated without cause. Today, at least, I'm aware enough of this tendency to catch myself before some minor frustration snowballs into outright anger. Back when I was 22, however….

ENTER SNOWBALL #1

Within Microsoft's unique corporate counter-culture, classical Dilbert-inspiring office policies are rarely to be found. The dress code is lax. There is little opposition to setting up a popcorn machine and charging supplies to the departmental budget. And with private offices with solid doors, no one can complain about your personal environmental preferences, save that of noise.

In Developer Support, we naturally followed suit (or the lack thereof). We rarely had any face-to-face contact with our customers and even then we impressed them so much with our enthusiasm and the quality of our work that outward appearances were somewhat unimportant (as told in the story at the end of Chapter Six). Having our own popcorn machine provided a quick snack and the ability to keep on working even if the cafeteria was closed or if we lacked pocket change for the vending machine. And it was a long-standing tradition within the cubicle farm of Developer Support that we generally kept the fluorescent overhead lights turned off. In the days before flat-panel LCD screens, programmers typically liked to work in moderately to dimly lit rooms because it's much easier to stare at a computer for twelve hours straight if there isn't any glare on the monitor. And without individual offices, we had agreed *en masse* to turn the overhead lights off and let everyone illu-

minate their own space with desk lamps. (For this reason our area was known as the Bat Cave.)

At the same time, upper management rightly wanted to encourage a certain degree of professionalism within our ranks because while we didn't *see* our customers that often, we *talked* to them all the time. The principle here is that one's demeanor is subtly influenced by one's environment in surprisingly powerful ways. As Malcolm Gladwell illustrates in *The Tipping Point,* if you are always surrounded by sloppiness, you will tend to express or favor sloppiness in other parts of your life; if you are surrounded by refinement, you will tend to express or favor refinement. This is true of how we hold our bodies, how we speak, how we write, and, indeed, how we even think. So be very careful in choosing your influences! Environment is generally stronger than will power.

So in the professional context of Developer Support, a threadbare T-shirt motif with an optional theme of 1970s-era running shorts, for example, wasn't exactly appropriate. Likewise, the popcorn machines were rather aromatic if not messy—besides the obvious fact that answering a phone while chewing a mouthful was *not* a very good idea. And somewhere it had been shown that people (at least those on a manufacturing floor) were more productive under bright lights. Therefore we were informed that thrift-store fashions could no longer be allowed, that the popcorn machines would have to go, and that the lights would remain on. Simple as that!

Now as far as most employees were concerned these things weren't even issues to begin with. They already dressed well, they didn't care one way or the other about popcorn, and a few of them actually preferred the overhead lights. A few of us, on the other hand, held the opinion—and voiced it fearlessly!—that these new rules were absurd. The slobs protested by dressing worse than ever. The popcorn junkies among us began to

employ the microwave variety with reckless abandon. And the Anti-Fluorescent League turned almost militant.

My personal thorn was the lighting. "We've been perfectly happy without the lights," I cursed under my breath, "so why should we have to keep them on now?" Failing to see any good reason for the new policy, then, I took it upon myself to keep the lights turned off. Whenever someone from upper management came by and turned them on, I waited until they left and turned them off again. When they had special tamper-resistant switches installed, and turned the lights on, I found a way to tamper with a paper clip and turned them off again. Above my own desk, I also found it convenient to simply reach up and pop the bulbs out of their sockets. But this was apparently a violation the local fire code, so someone kept having the bulbs put back in. I pulled them out again anyway.

Obviously this sort of tussle didn't have the most uplifting effect on my attitude. It was, in fact, degenerating a little more every day. "Why," I fumed, "can't we just be left alone to do what we're supposed to be doing? Why does the lighting have to become such an issue?"

I wasn't helped by the fact that at this time I was deep into William Shirer's comprehensive history of Nazi Germany, *The Rise and Fall of the Third Reich*. What did I just say about choosing your influences? Reading that book heightened my sensitivity—and my stubborn resistance—to all forms of what I considered petty tyranny, whether real or imagined. I also read the classic team-management book called *Peopleware* by Tom DeMarco and Timothy Lister, a wonderful title that I highly recommend to anyone in a leadership position. If, on the other hand, you're an unempowered cube-jockey like I was, better give it a miss. For me, reading that book simple highlighted everything I felt our leaders *weren't* doing right, and that didn't help my attitude at all….

ENTER SNOWBALL #2

During Microsoft's first fifteen years or so, it had pretty much been the case that once a development team sent a product to manufacturing it became the responsibility of Product Support. If there were serious problems, well…of course the development team could provide a patch or an update. But if end-users were having trouble installing it, running it, or simply understanding it? That's what Product Support was for.

As products became more complex, however, they were also becoming more and more difficult—and expensive—to support. At the behest of the development teams themselves (and over the protests of our own upper managers, as I learned later), it was decided somewhere on high that Product Support should submit a quarterly operations cost to each product development team who would then reimburse the expense. This would give those teams a reasonable measure of how "supportable" their products were and make them directly accountable for it. They would then naturally want to create better products and thus reduce their financial liability. So went the theory.

In practice—to generate the cost figure—someone decided that all engineers within the Product Support Division would report, accurate to the quarter hour, how much time they spent each week supporting different products. A simple system was devised to categorize those numbers by product, add them up, and send the bill to the appropriate development team.

For most of Product Support this process was practically effortless: 95% or more of the engineers only ever dealt with a single product. All they needed to do was report the time they spent answering customer questions, save a bit here and there for the occasional excursion into foreign territory or the time they spent in research and ongoing education.

In the Premier team where I worked, however, it wasn't nearly so simple. We (all four of us) personally and individually handled each and every question from our specific clients no matter what those questions involved. Solutions often crossed the boundaries of two or more products, sometimes as many as five or six. I could just imagine:

Me: "Thank you for calling Microsoft Developer Support. How can I help you?"

Caller: "I'm trying to connect to a SQL server from within our Windows application, and…"

Me: "OK, hold on a minute, let me record this. <mumbling> Five seconds for the Windows Software Development Kit, start clock for SQL Server…OK, please proceed."

Caller: "Uh, yes…this SQL server is running as a background process under OS/2…"

Me: "Got it…ten seconds for SQL. What's next?"

Caller: "Huh? Er…well…this OS/2 machine has LAN Manager 2.1 installed whereas the Windows client is running Novell NetWare 6.3 and…"

Me: "…OK, OS/2, eight seconds, LAN Manager, five seconds, Windows, four seconds, Novell NetWare…wait a minute, we don't support Novell…how do I count my time for that?"

Caller: "W-what? Why are you counting seconds?"

Me: "It's my job, don't you know?"

Caller: "Uh…"

I'm exaggerating, of course, but you get the idea—it seemed ridiculous to expect that our numbers would mean anything

unless we expended gobs of energy just to record them. Even if we did, our supposedly accurate numbers would ultimately be collapsed with hundreds of others into a single figure on someone's budget spreadsheet for apparently no other purpose than shuffling some money around.

As a team, then, we decided it would be much easier if we just made some numbers up. To spare us even that trouble, my teammate Charlie Kindel whipped up a little random-number program to do all automatically!

Well, word got out about our little trick and someone was sent down to put things straight. Standing in my cubicle, the fellow who was second-in-command in Developer Support—a man I'll refer to as Arnold—did his best to give us, and myself in particular, a real pep talk. With admirable enthusiasm and a transparent belief in the goodness of the whole system, he patiently explained why our participation was important for our department, our customers, and Microsoft as a whole.

Now, had I been more in control of myself I could have just nodded along with Arnold while continuing to have Charlie run numbers for me. But *noooo*…already perturbed about those cursed fluorescent lights (which Arnold had just turned on again, if I remember correctly), I wasn't in the mood to play sandbag to what I saw as another outrageously cumbersome policy. Allowing my red-haired temper to get the better of me, I gave Arnold a piece of my mind. "The whole scheme is preposterous!" I snarled, "It's not worth the effort to track my time. So whether you like it or not, I'm just going to make my numbers up." In fact, I told him that he could save me a lot of trouble by just making them up for me. So there! Ha! Grrr!

I don't ever remember being quite so angry with any manager as I was then. Nor had I probably ever been so close to getting myself fired! But I just couldn't help it. From my point of view I was doing everything possible to help our customers,

while upper management seemed bent on obstruction. They seemed far more concerned about how *long* I spent answering questions rather than how *well* I answered them, not to mention whether I was answering them in a brightly-lit cubicle!

"Well," I consoled myself with a sigh, "at *least* they aren't rating our performance on the *number* of questions we answer. That would really be silly…."

ENTER SNOWBALL #3

The 1990 introduction of Windows 3.0, along with a number of other stunning products like Word for Windows, created a dire crisis in Product Support. Sales were skyrocketing and millions of new users were suddenly calling in for help. Caught off guard by this unanticipated boom, end-user Product Support was utterly overwhelmed and desperately understaffed. Callers invariably had to wait in a queue, listening to Microsoft promos for as long as an hour before finally talking to an edgy support engineer who had already spent upwards to seven hours on the phone that day answering question after ceaseless question. Phew!

Under these intense conditions, the overriding concern was to answer, in some way, as many calls as possible and to minimize the wait time lest callers became disgruntled and dissatisfied. It was perfectly reasonable to then rate the performance of each engineer according to the number of calls answered in a day. Such a measure would more or less reflect one's ability to hone in on the customer's core question, get it answered, and quickly move on to the next call. While this emphasis did tend to undermine quality a bit, it was simply preferable to give *some* kind of answer than no answer at all.

Goals were therefore set to increase the number of calls answered, keep the wait time under two minutes, keep the average call duration under four minutes, and to prevent customers from hanging up while waiting in the queue. To monitor the ongoing battle, automatic tracking systems were installed that produced copious real-time statistics. Team managers could then rally the troops whenever an extra burst of energy was necessary on the battle front.

It wasn't long before all this proved marvelously effective. Calls were getting answered, fewer customers were dropping out, and the whole situation became manageable. What's more, the support engineers responded well—meeting and in many cases exceeding their numerical goals.

In Developer Support we also felt the surge as many new software companies were now jumping on the Windows 3.0 bandwagon. But whereas the sale of a million copies of Windows created a million potential callers to end-user support, it only meant another couple of hundred or so in our neck of the woods. Spread around three or four dozen support engineers, the actual increase wasn't all that significant. Which was good! Unlike many end-user questions that could be answered in a few minutes, programmer questions could take several hours; the ones we dealt with in the Premier team often took days, even weeks. What mattered for us was that we got the questions answered thoroughly and accurately, and that we helped programmers understand the system more deeply. The number of questions we actually answered each day was completely irrelevant as long as we weren't falling behind the incoming requests.

Nevertheless, word came down from on high that we would also now be rated, like everyone else in Product Support, by the daily number of phone calls and service requests we answered: no more, no less.

"My God," I exclaimed, (embellished with my now customary under-the-breath profanity), "doesn't upper management understand what we actually *do*?" Here we were, especially in the Premier team, trying to help some very large companies move onto the Windows platform. Here we were, trying to understand the most intricate parts of a complex operating system. And here we were, trying to share information that would *reduce* the need for programmers to call us in the first place! "How on earth," we asked ourselves, "can they rate our performance on volume alone? Why, wouldn't we then be better off spreading *mis*-information?!"

In truth, our upper managers actually balked at subjecting us to this sort of thing; they really weren't given much choice themselves. And, as it turned out, such measures never became very important in our department—despite anything we heard to the contrary, our performance continued to be measured in terms of quality as well as creativity. One more incident, however, prevented me from really understanding any of this, an incident that made me feel that everything I was working for was being sacrificed on the altar of mindless statistics.

In early August 1991 my whole team took two days off to attend the Windows 3.1 Developer's Conference in downtown Seattle. We got a jump-start on the yet-to-be-released operating system and were able, for a change, to meet many of our clients face-to-face (we did dress decently). In our absence, though, we accumulated a sizeable backlog of service requests and phone messages. What's more, those customers who also attended the conference were bursting with new questions themselves, and called in as soon as they returned to work.

That next day we were thus faced with both the backlog *and* a call volume at least five times the norm. If there is a special place in the 666 layers of Hades reserved for support engineers, this was it. A normal day for me meant answering

maybe five calls and maybe three electronic service requests. This day—throwing precision, as necessary, to the wind—I answered a combined total of over sixty. After eight-plus nonstop hours, I was completely fried. Serious ultra-crispy.

It was at this exact point that chummy ol' Arnold stopped by my cubicle. Was it to talk about the lights? Thank God, no. Was it to talk about time-tracking numbers (which Charlie's program was still generating for us)? Nope. Was it to console me? Not. He actually came by to congratulate me. Yes, *congratulate* me. Having seen a big spike by my name on the call-tracking monitors, he enthused over how wonderful a day it must have been for me. "Wonderful?" I thought, "this one most *miserable* day of my entire career?" I really don't know what he was thinking. As for myself, I wanted to slug him. But exhausted as I was, I could only keep myself from weeping….

Thus it was a week or so later, in my performance review with Dan Quigley, that I stated with unequivocal certainty, "ten of me would *not* work here." I was thoroughly disgusted with the whole mess. All I wanted was the freedom to help programmers in every way I could imagine. Why couldn't I do this in Developer Support, of all places? Why? *Why*? WHY? I felt crushed on all sides and didn't know how much longer I could stand it. Something had to change or I would just explode.

Of course, I felt that the whole problem was with Developer Support's upper management. "They're the ones," I fumed, "who are the problem!" From the more unbiased perspective of over a decade, however, I see the error in that judgment. All the rules, all the policies, and all the tracking systems were perfectly acceptable and even desirable to the vast majority of support engineers. Many of my co-workers were happy then and continued to work happily in the group for years to come.

At the same time, my own feelings were somewhat justified by the fact that I was not alone in them—a fair number of

others felt the same way, even if their experiences were not quite as intense. What's more, I couldn't believe for a second that my compassionate desire to help other programmers in every way imaginable was at fault. Sincere kindness and love is never a mistake.

But attachment is. It prevents us from seeing what's really trying to happen in any given situation. It prevents us from considering other possibilities, or other environments, through which we might better exercise our natural impulse toward growth and realize a greater degree of self-expansion. Being out of tune with this greater harmony, we experience pain.

In my case, my sympathies were expanding beyond the boundaries of Developer Support's limited though altogether valid mission. At the same time, I clung to the mistaken belief that Developer Support was the one and only place where I could express those sympathies. Consequently, I kept painfully smashing into walls no matter which direction I tried to go, and in my frustration I could only blame the walls themselves or the people who had put them there.

To put it more gently, helping other programmers succeed was my special passion. From the beginning of my full-time career Developer Support had been the soil in which that love, like a plant, could flourish. Eventually, however, the roots of that plant became bound, constrained, and even starved for nutrients to the point where better soil—or different lighting!—could no longer help. In short, what that plant needed to grow further and ultimately flower was simply a bigger pot!

Why, then, did I have to experience so much frustration in the process? Well, you might remember from Chapter Three that young computer engineers like myself were generally not all that interested with making their careers in the relatively inglorious fields of product support or marketing. No, we all wanted to be on the front-lines of high-tech innovation! For me,

having failed at becoming a hot-shot software engineer, the opportunities I had in Developer Support seemed the next best thing. And had I not been so discontent with my circumstances, I would have stayed in that group for a long time, comfortable and complacent.

Complacency is perhaps the greatest obstacle to growth of any kind, especially spiritual growth, and it was exactly this complacency that God was now helping me avoid. Being open and expansive is the real solution for difficult situations like this. By allowing me—even inspiring me!—to become so upset with Developer Support, God gave me the openness and the courage to accept just about any other opportunity he might place before me, especially one that I would certainly have rejected outright under better conditions.

And what was that opportunity? You got it: marketing!

Shortly after my review with Dan Quigley, I got a call from my old friend and former manager, Bob Taniguchi. "Hey Kraig," he asked, "how would you like to work for me again?"

How would I!

Bob was then working in the recently formed Developer Relations Group (DRG), a kind of technical marketing team that sold faith rather than products. In particular, they promoted the newest technologies in Microsoft's yet-to-be-released operating systems—the so-called Gospel of Windows—to other software development companies. Bob was recruiting me to give DRG a more solid technical grounding and a better rapport with outside programmers. They needed someone who could pro-actively remove whatever technical barriers (poor documentation, lack of sample code, and so on) might prevent the adoption of our latest innovations. DRG also wanted an internal advocate for the developer community at large.

How could I turn it down? The job was virtually tailor-made for me. It meant working with the cutting edge of new technol-

ogy as well as serving other programmers more directly, more creatively, and more effectively than Developer Support could possibly accommodate. It was perfect.

Still, I had to wonder just a little bit. Would I again run into boundaries? Could I truly give it my all and not wind up with ulcers? Would Developer Relations really give me the freedom I had been seeking in Developer Support?

That question was put to rest forever when I visited the man—whom I will call Friedrich—who would become my direct manager if I accepted the position. "The purpose of this job," he said, summing up the deal, "is to make yourself obsolete." The only measure of this goal would be my willingness and effort to give away absolutely everything I knew and learned. Music to my ears!

As a token of my acceptance into his group, he handed me a 40-ounce Louisville Slugger, the biggest and heaviest baseball bat you can find. (The significance of this will be explained momentarily.) For the next two months or so, as I finished up my work in Developer Support, I often carried this bat around on my shoulders. It was way better than having one of those little plastic pyramids on my desk. For rather than merely symbolizing past accomplishments, it symbolized the strength of my convictions, the height of my ideals, and the future fulfillment of my aspirations.

And it also seemed to prevent anyone from confronting me when, as I continued to do until the day I left for DRG, I once again walked over to the switches, and turned off the lights.

CHAPTER EIGHT

A Mile in Their Shoes

May Thy love shine forever
On the sanctuary of my devotion,
And may I be able to awaken Thy love
In all hearts.

–Paramhansa Yogananda

"Vee play HAAAARDBALL hier!"

During our first meeting to discuss my joining his team, my would-be manager, Friedrich, was explaining in his characteristic German accent how Microsoft's Developer Relations Group generally did business. DRG didn't like to take no for an answer: its Technical Evangelists, as they were called, were out to spread the Gospel of Windows—even if it required certain forms of, shall we say, intimidation. To make his point symbolically clear, Friedrich held aloft the 40-ounce Louisville Slugger that he subsequently gave to me.

I wasn't quite sure how I would apply this approach to my particular role. Coercion just isn't part of my nature. If I was to succeed in my new position with DRG (where, ironically, I now had the title of "software design engineer"), I'd need to learn how to win others to our systems without bullying them. My goal was to make the task of making Microsoft's newest tech-

nologies easier for programmers to understand and implement: I simply couldn't do that by force!

On a brisk Monday morning, the 28th of October, 1991, I came in bright and early to my new office in Building 9 at Microsoft's corporate campus in Redmond, Washington. Fifteen months earlier, with a heavy heart, my belongings had been moved away to Developer Support, located in nearby Bellevue. Now, with a heart renewed and uplifted, my belongings had been moved back.

Or so I thought. On the Monday morning after a move, you normally expected to have certain things in your office like your packing boxes and, you know, furniture. Mine was devoid of all solid matter save the secretarial return of a desk standing on end. The walls were bare. The phone jacks were dead. A few loose network cables dangled from the ceiling. And my boxes? They were still six miles and eight floors away, waiting patiently in the downtown Bellevue high-rise that had been my home the previous Friday.

It was not the most auspicious way to begin a new job! But DRG's exceptionally capable administrative assistant managed by day's end to procure a full desk, a working phone, a whiteboard, a terminal through which I could check my email, and a guarantee that my boxes and computers would show up the next day. By Wednesday I was running full steam ahead.

The Developer Relations Group originally formed in the late 1980s after the first version of the highly-touted OS/2 operating system turned out to be a market flop. Not that it wasn't a good system in and of itself: at the time it was clearly technologically superior to MS-DOS and Windows in just about every way imaginable, including its big selling point, "pre-emptive multi-

tasking."* Still, it didn't sell.

Various folks in Microsoft's Systems Division got together to find out why. Technological superiority, they discovered, appealed only to die-hard geeks who bought the system on that principle alone. Everyone else, on the other hand, didn't even pretend to understand this pre-emptive multitasking thing let alone why they should shell out their hard-earned greenbacks for it. All they wanted were innovative OS/2 applications that took special advantage of the operating system to help them get their work done better and faster.

From this perspective it was easy to see the failure: the few dedicated OS/2 programs that were on the market, like one recipe-management database system, were about as interesting as oatmeal. Better software was available for even long-obsolete versions of MS-DOS. So if the new operating system was to ever get out of the kitchen, so to speak, Microsoft had to get Independent Software Vendors† (ISVs) excited enough to create truly innovative applications for it.

Thus was born the Gospel of OS/2, which went something like this: "The new technologies in OS/2 are *way* cool. If you, our wonderful, *friendly* ISVs take full advantage of these technologies, then *your* products will be *way* cool. Customers will *love* them. They'll *clamor* for them. You will sell *more products*. You will make *more money*. This is goodness." And from Microsoft's standpoint, of course, sales of the requisite operating system would naturally follow and the value of everyone's stock options would naturally increase. This, also, was serious goodness.

---

* A system feature that allows multiple programs to apparently run simultaneously, even if they were written to run exclusively.

† A term used to refer to any software company outside Microsoft.

To disseminate the Gospel far and wide, Microsoft then gathered together a handful of choice zealots—including Bob Taniguchi—and formed the Developer Relations Group. "And YEAH," sayeth the chronicles, "each bearing thus the distinctive and holy mark of Technical Evangelist, the chosen few WERE sent forth to spreadeth the Good News to all who wouldeth listen. And YEAH, if they DIDETH not listen, or they dideth AND yet faileth to surrender theirs hearts to the Gospel…then playeth hardball, did they, the few…"

Well, many ISVs did listen and a good number took action. Month after month, the Gospel of OS/2 steadily gained new converts. Then came The Great Schism (see Chapter Five, page 64): IBM took control of OS/2 and left DRG with little more than disillusionment. But it didn't last for long: just as Windows NT arose Phoenix-like from the ashes of Microsoft's OS/2 efforts, so did the Gospel. All it needed was a little revision. Overnight, the Gospel of OS/2 became the Gospel of Windows.

Now until this time Windows wasn't much to shout about. Each of its first three incarnations offered little more to ISVs than basic services; only the largest companies—the ones with lots of excess R&D capital—could offer truly innovative features to customers. Smaller companies, unable to develop such enhancements themselves, could only compete with each other for the market leftovers.

Windows NT and Windows 3.1, however, were changing the story dramatically. These systems offered such a powerful collection of new built-in technologies that small ISVs suddenly found themselves on a level playing field with the big guys. In fact, they had the advantage. It was no longer a matter of who had the most money but who could respond most quickly—something at which smaller companies normally excel—with applications that took advantage of those technologies. And those who did would ride the great wave of prosperity.

By this virtue alone, ISVs were eager—even impatient!—to hear the transfigured Gospel. They flocked to DRG's developer conferences. They devoured DRG's white papers and press releases. They begged and pleaded DRG to bestow a blessed visitation on their corporate headquarters, even offering to pay everyone's expenses if that would help! Nobody, it seemed, wanted to miss out.

With demand rising exponentially, many more Technical Evangelists were needed. Every other week a new zealot was hired, indoctrinated, and sent out to preach The Word. In less than a year the ranks of the DRG apostolate had swollen from an elect few to a vibrant thirty-five.

In the process, of course, people—imperfect as they are—made their mistakes. Patience and sensitivity seldom flower in the garden of religious fervor; the weeds of boisterous dogmatism choke out the more unassuming buds of honesty and sincerity. Truth is easily forgotten in the desire to gain new converts as quickly as possible.

In the case of Developer Relations, the not-so-subtle distinction between *imposition* and *sharing* was sometimes lost on those Evangelists whom Bob colorfully described as "technical arm-twisters." In their fervid crusading to please He-Whose-Power-It-Is-To-Grant-Stock-Options, they forgot that marketing is a function of friendship, not force, and one that *invites* rather than commands. Blinded by passion, they played their hardball to a fault, resorting, on occasion, to such questionable tactics that they succeeded only in creating alienation and distrust. And the trend was only getting worse.

This was the Developer Relations Group into which I came. DRG wanted to re-establish and strengthen its rapport with outside developers and for this it needed a fresh infusion of genuine compassion. Bob sought me out because he knew that my time in Developer Support had taught me both selfless

service and habitual sincerity. I just couldn't allow myself to put anything above a programmer's individual needs, be it personal desire, group pressure, or even Microsoft's bottom line. Indeed, in Developer Support I had often given special knowledge to some of Microsoft's fiercest competitors that our own development teams didn't have simply because they'd never asked for it! And by doing so I gained the appreciation and trust of at least a few individuals in those other companies. Now in Developer Relations I would do the same thing on a much larger and much more visible scale.

My first assignment in this new role set a good precedent. Windows 3.1 had some new "localization" features that were intended to greatly simplify the process of adapting software to overseas markets. A few weeks before I'd transferred to DRG, Friedrich had asked me to see if they truly were simple and to understand how ISVs could use them most effectively. He gave me a good incentive, too: at the end of my very first week with the group I was to fly to Scottsdale, Arizona, with a few of our evangelists where we'd been invited to give presentations on Windows 3.1 to Symantec Corporation's annual engineering retreat.

Technically speaking I was fully prepared to give my half-hour presentation on the details of localization. But that was the *only* way I was prepared! When my turn came, I found myself staring out into a vast, dimly lit sea of two hundred eager and information-hungry nerds. Four hundred eyes (eight hundred if you count the glasses) were fixed on me, and four hundred ears on the wisdom I was expected to impart.

I was suddenly more nervous than I knew possible. Had I been able to flee I most certainly would have, but with nowhere to run I chose the only logical alternative (other than sudden cardiac arrest): get the talk over with as *quickly* as possible! From the moment I opened my mouth I was pegged on fast-

forward. And after completely running myself out of breath several times—blahblahblahblahblahblahbla…**gasp** I-N-H-A-L-E! blahblahblahblah…—I suddenly realized that my half-hour talk had taken all of twelve minutes. Obviously I wasn't a natural at this sort of thing.

Stage fright notwithstanding, I was at least sincere. I had no interest in preaching some sermon with the thought "this is the truth and you must accept it." Instead, I completely focused on the needs of my audience, explaining (rather quickly!) which localization features could be wholly ignored under a variety of conditions and offering ideas for using certain features in somewhat unorthodox, albeit helpful ways.

That evening, as we schmoozed with Symantec's developers in our hotel's outdoor hot-tubs, a number of them expressed appreciation (as well as amusement) for my effort. They were convinced that I had their best interests in mind and wasn't simply some glazed-over Microsoft poster-boy. They felt as though I had stepped into their shoes and understood their particular reality.

On my return to Microsoft it occurred to me that this kind of empathy might actually be the most effective approach to my work. Service is rooted in sincerity and sincerity is rooted in the love of truth. To see truth one must step outside the delusive limitations of any one perspective and relate meaningfully to other points of view.

As if in direct response to these thoughts, my next assignment afforded the opportunity to deeply explore this idea: not only did I get to step into the shoes of other programmers, I got to walk a mile in them! Indeed, the journey that began here covered many miles, as this one task became the cornerstone of my remaining career at Microsoft.

I was given the task of understanding and explaining a new technology called Object Linking & Embedding, or "OLE" for

short.* OLE was an outgrowth of the user interface metaphor known as the "clipboard" and a cross-program communication mechanism called Dynamic Data Exchange (DDE). Both had been around since Windows version 1.0 to allow applications to share information with each other, such as text and graphics. By adhering to certain standards and protocols any program could exchange data with any other, provided they both understood the data formats and protocols in question.

In the late 1980s there was a proliferation of new data formats as audio, true-color graphics, and video all came of age. What's more, the need to integrate different applications for the creation of rich "compound documents" with such elements was very much on the rise. Yet it was increasingly costly and difficult to update programs to understand every new custom design for achieving these ends. Programmers themselves were getting more and more fed up with the whole mess.

So Microsoft and a number of key ISVs banded together to create a somewhat revolutionary solution: all new data formats would be wrapped inside a standardized abstraction. Internally, these "objects," as they were called, would contain both specialized data (image bits, color tables, video frames, etc.) and the necessary "intelligence" (program code) to manipulate that data and present it to the end-user. Programmatically, however, these objects would all interact through a standardized interface. This meant that any program capable of hosting the abstract "object" could incorporate all kinds of new data formats it otherwise knew nothing about, as well as those that had yet to be invented. Likewise, any program that provided such objects could be used with all present and future hosting programs. Such was OLE.

---

* Pronounced "oh-LAY," as in the Spanish *olé* (sans matador).

Of course, I'm giving you a simplified picture—the OLE standard was making it possible to do extraordinary things and, as a result, it was extraordinarily complex. Many intricate operations were required to create or manipulate an object. The protocol was also fragile: any mistake along the way spelled disaster. In effect, OLE was a new language that programs and programmers would have to learn to speak perfectly lest the whole thing collapse.

Offhand this didn't bode well for the technology's success. The work required to make a program OLE-compatible was as much, if not more, than making a program understand a veritable pantheon of different data formats. Fortunately, a good number of "sentences" in this new language were so common that they begged for standardization themselves. That is, many series of common tasks could be implemented as shared code libraries that the operating system would make available to all applications. This, in and of itself, would relieve around 80% of the burden.

As the operating systems vendor, Microsoft formed a small development team to deliver the goods. A high-level software architect designed the final technology and refined its protocols; two software engineers created the libraries; two technical writers documented everything; and one other engineer both tested the libraries and assembled a few sample programs. That was pretty much it.

Developer Relations was then asked, as usual, to spread the technology among ISVs, and spread it they did. By the time I joined DRG, our evangelists included something about OLE in nearly every presentation; advance copies of the OLE team's work were flying out the door. And having given it such importance, ISVs were heeding our message and jumping on board, hoping with an almost blind faith that OLE would lead them to glory and riches.

Unfortunately, the Book of OLE in the Gospel of Windows wasn't very well developed; few of our evangelists really understood it. While they preached the "good news" everywhere they went, they failed to point out that the technology was not universally applicable—its particular focus made it wholly inappropriate for many types of programs. To make matters worse, most ISVs couldn't make enough sense out of the thing to see this for themselves: the specifications, they said, were cryptic; the documentation unreadable. Before long, ISVs complained about the time they were wasting with this infant technology. Moreover, they were starting to distrust DRG for their "misleading" messages. Not good. Not good at all!

So in the latter part of November 1991 Friedrich asked me to tackle the problem. There were obvious barriers to the widespread acceptance of OLE: could I find and eradicate them? Unaware of what I was getting into and where it would lead, I gladly accepted the challenge.

With my empathetic approach in mind, I decided to effectively *become* an ISV and undergo their plight for myself. With nothing more than the materials they had received from us I sat down to incorporate OLE into a few of my own programs. Mentally I isolated myself from the rest of Microsoft: I vowed to struggle with questions to the best of my ability before calling on the OLE team for answers. I was even physically isolated at the time—for one thing, my office was temporarily separate from the rest of DRG (in Building 5). And thanks to the never-ending Microsoft Shuffle I was, for six weeks, only one of about eight people left in all of Building 9.

This was an excellent thing, as it turned out, because once I started to feel frustrated and confused I often announced my feelings quite vocally!

It wasn't that the shared code libraries weren't good—they were wonderfully fulfilling their purpose and the OLE team

was apparently doing a fabulous job of getting them written, tested, and documented. However, it turned out that the whole project was far more complicated than the team had imagined. Under intense time pressure the technical writers could only describe *what* the libraries did but not *why, when,* and *how* to use them. The test engineer, under similar pressure himself, was pretty much forced to turn his sample programs—which would normally answer such questions—into complicated test suites for the sake of getting the libraries debugged.

This left each individual ISV with some astoundingly difficult and practically insoluble problems. Facing this for myself in virtual isolation I not only understood why ISVs were feeling upset with Microsoft but truly felt their pain. In fact, I literally became angry at the company—as if I were an outsider—for heaping such an unthinkable burden on me! Had you walked by my office at the time you might have been, shall we say, blessed with one of my many colorful litanies and rather irreverent invocations of, shall we say, a certain Jewish chap who taught among the hills of Galilee about twenty centuries ago. Everything was so confusing that I felt as though I'd been pounding my head into a brick wall until my upper skull developed a rather sporting plateau.

The whole time, however, a part of me stood back and observed. I had deliberately chosen this path out of love and compassion. My desire to help other programmers wasn't just a job: it was a mission. I willingly chose to take great pains upon myself in order to free hundreds, perhaps thousands of others, of the agony I was experiencing. In a sense, I was acting out in my own insignificant way—though it may be equally irreverent to say so—the Passion of that same Jewish fellow that I'd been invoking so often. And in this general thought I found the necessary inspiration to keep me going through the greatest of trials.

Then something shifted. Something within me stirred… something that went beyond mere compassion. As I finally began to break through the barriers—that is, when I finally got my own damn programs to work!—I was experiencing more than just relief. I found myself experiencing…joy.

Yes, *joy*. Perhaps you don't think that getting a program to work could be all that joyful. But it is to a programmer. Really. You see, many of us types got started by writing useless little programs that did something "really cool" like bouncing a ball around the screen. For us, watching a bunch of symbolic source code generate something tangible was utterly magical and utterly joyful. Then we went to college, became professionals, and suddenly had to do "real work," earnest and demanding. No more horsing around: we were being paid for it now and had to take our work seriously. And if that meant toil and tears, as I had so often witnessed in others and as I had just experienced in the midst of OLE, so be it.

But now, after much pain and anguish, I had somehow emerged on the other side of this "serious" business. Again, I wasn't just feeling relief, as is so often the case when a burden is finally lifted. No, I realized that I was once again feeling that joy of bygone days. I was actually having fun. Real, honest-to-God *fun!*

Pausing to reflect upon this deeply refreshing experience, I realized that professional software engineers were, for the most part, usually under such immediate deadlines and other pressures that "fun" just wasn't in the equation. Somehow in the process of earning a living we'd forgotten the magnetic joy that had first attracted us to programming.

In that moment my deep desire to help other programmers took a new form. I didn't just want to help them get things done: I wanted to help them *enjoy* it. I didn't simply want to relieve their pain: I wanted to *inspire* them. And I didn't want

to merely educate them: I wanted to *awaken* them to new possibilities and potentials....

Yes, that was it! I felt joy and I wanted to share it! And by sharing it, I wanted more than anything to awaken that joy in others. This, I realized, was my real mission at Microsoft, even, in a sense, my personal *ministry*. My position within Developer Relations, my responsibilities, the very technologies I worked with—everything!—these were vehicles not for The Gospel, but for sharing joy. After all, joy is what we are all really seeking in every activity. To lovingly share one's inner light and awaken joy in the hearts of others is perhaps our highest outward responsibility as human beings.

Deeply inspired by this newfound purpose—though at the time I still didn't think of it in such overtly spiritual terms—I immediately began to apply it to my work with OLE. I wrote, for instance, a pair of "recipe" books that walked programmers through a step-by-step process of incorporating the technology into a program. But no dry technical manuals these—while making sure they were both accurate and complete, I also tried to make them interesting, fun, even humorous on occasion. And I must have succeeded—people repeatedly told me that they were "a real pleasure" to read. The sparkle in their eyes told me that I had touched them in a meaningful way, even if they didn't say so directly.*

My sample programs too, proved equally touching. People told me that they read more like documentation than code

---

* An Acquisitions Editor for Microsoft Press liked them well enough that he asked if he might just slap covers on them and sell them retail. While they didn't quite have the necessary quality for such treatment, they later appeared as part of the *OLE Programmer's Reference* from Microsoft Press. If you are interested in taking a peek at the originals, they can still be found on msdn.microsoft.com by searching for "OCLIENT.EXE" or article 81198.

(which easily becomes gibberish even to programmers). Not only did they consider them the most useful samples they had ever seen, but also felt a satisfying sense of discovery as they worked through them and understood the technology more and more deeply.

How can I express my inner gratification? I delighted not in the personal praise or even in the work I'd done but rather in seeing people *enjoying* their experience with OLE. Truly, it was more than I had ever hoped for.

Indeed, as time went on an even deeper effect of this work began to reveal itself. You see, as these little products of mine made their way into the hands of many ISVs, I quickly became, to the public eye, the recognized "OLE expert." As such I was invited to give detailed presentations at some of the most widely attended conferences in the software industry.

After these lectures people usually plied me with questions and asked for specific advice on incorporating OLE into their projects. In the process of giving the latter, especially, I was pleasantly surprised to find myself not just thinking up a number of creative ideas on the spot but also encouraging each person to see a higher potential in both their work *and* themselves. In effect, I had somehow managed—almost without effort—to turn a potentially gruesome chore into a kind of exciting adventure in software engineering—"software" both for the computer and for the human being!

As I saw this sense of adventure come alive in people's faces, I realized that though many of my creative ideas would not prove the least bit feasible, it really didn't matter. I knew that people's kindled enthusiasms would eventually help them find their own path, and their own answers. And I knew that this, more than anything, perhaps, would create a vibrant and dynamic software market in which Microsoft's own products would flourish.

I didn't have to preach.

I didn't have to proselytize.

All I had to do was share my joy,

and people would convert themselves.

CHAPTER NINE

Only So High

"Windows! Windows! Windows!" Microsoft has been accused of many things in its history but it can never be accused of timid marketing. Thanks to Steve Ballmer's energetic and unrelenting enthusiasm for the company's flagship product, Microsoft's marketing schemes are the stuff of legend—grist for both business school seminars and lawsuits alike. And thanks perhaps to the fact that every new Microsoft employee was shown, during their first day's orientation, a video of Steve pushing Windows in the manner of a used-car jockey or second-rate appliance dealer—plaid jacket and all!—it comes as no surprise that Microsoft's style has occasionally challenged the limits of convention.

For example, there was a rather jovial technical evangelist in Developer Relations by the name of James Plamondon. His specific focus was wooing Apple Macintosh programmers to instead concentrate their energies on Microsoft Windows. The task was difficult, considered virtually impossible by some: Macintosh ISVs were a particularly loyal bunch, a sentiment repeatedly reinforced by lavish attentions from Apple itself. Many Mac programmers were openly hostile to Microsoft and anything to do with Windows. Anyone from Microsoft's System

Division who had the audacity to attend a Macintosh developer's conference most certainly played the part of Daniel in the lion's den!

Nevertheless, prior to the release of Windows 95, James the Imperturbable regularly did so, fearlessly willing to become a martyr for the cause. Fortunate for him, perhaps, Apple was deeply into corporate restructuring at the time. Resources were thin; layoffs plentiful. Consequently, the flow of perks from Apple's evangelism department to the Macintosh programmer community had tapered off considerably. To worsen matters, Apple apparently hadn't responded well to developer needs in their latest version of the Macintosh operating environment, System 7.

There was a subtle buzz among the programming proletariat: "System 7 sucks!" was the way they put it. And though the next update (System 7.5) was better, it still didn't inspire widespread cheer.

For St. James the Opportunist, these were moments not to be missed. At one event called MacHack '94, James sponsored a massive impromptu pizza feed for all 300 attendees. This won him great acclaim, even from some of Apple's own employees. It also got him a nomination for the show's "Best Hack" award and some free publicity for Microsoft in *MacTech* magazine. At another event, MacWorld '95, James organized free day-long seminars on "Windows Programming for Mac Developers." He hired two well-known Macintosh developers for the job, shamelessly promoted the seminars, and printed up, as rewards for completing the seminar, simple T-shirts that read (in a blatant rip-off of T-shirts that Apple had made) "Windows 95 Sucks Less." And though he went about 500% over budget on all this, his contribution to the world of fashion at least won the Best T-Shirt of 1995 award from Computer World, a weekly trade magazine.

On a larger scale from James' legendary exploits, we also have Microsoft's epic product launches. The launch of Windows 95 was a kind of *coup de grâce* in this regard, achieving a degree of global awareness heretofore unrealized. Somehow Microsoft managed to make it really cool for any person-in-the-street to know about the thing—and desire it beyond reason. People you never imagined could possibly care about an operating system suddenly wore the mantle of "nerd" with the best, the brightest, and the geekiest. Reporters everywhere talked it up for months. Many people bought computers for the first time for the sole privilege of being in with the crowd. And right before the launch, a New Zealand man camped out for days in front of a computer store for the distinct honor of being the first person in the world to cough up a hundred bucks for his own personal copy!

The origins of this sweeping campaign style, and my first direct experience with a product launch, came more than three years earlier in May 1992. Chicago! Nearly all of Developer Relations flew *en masse* to Windows World '92, run that year in conjunction with Spring Comdex; total attendance to exceed 100,000. Here Microsoft launched Windows 3.1, the first version of the system to really establish itself in the marketplace and the one that set the stage for Windows 95.*

To prepare the hearts and minds of the public, DRG's evangelists dreamt up amazing ways to paint the town "Windows." For starters, they got almost every software retailer in Chicago to display the new product in their storefronts on the day of the launch. Then they managed to get flags bearing the Windows

* The one intervening launch was for Windows for Workgroups 3.11; this was, as mentioned in Chapter Six, done up on Broadway as a musical comedy. Bill Gates has never danced on stage since.

logo hoisted above downtown landmarks like Soldier Field, the Museum of Science and Industry, and even, if memory serves, the Sears Tower. They even had a big Microsoft Mouse walking all around O'Hare airport and the meals on every American Airlines flight into town that week included, compliments of Microsoft, a miniature Windows 3.1 product box with several Hershey kisses inside. In short, the city—and the sky above—was alive with Windows![*]

DRG was equally energetic inside the convention hall. I was one of the few, the proud, and the brightly attired chaps who got shanghaied into a rather brazen scheme prior to the opening of the show. Trotting around in obnoxiously loud neon-yellow "Ask Me About Windows" T-shirts and burdened with comparably obnoxious neon-green duffel bags full of marketing schmaltz, we set out to prove the "seamless compatibility" of Windows 3.1. Microsoft had promised that if you installed the new system on any machine running Windows 3.0, everything—yes, everything, old and new alike—would still work. To prove it, we tried to convince other exhibitors then and there to install the new system on their demo machines from the floppy disks we carried for this purpose.

Now since you've probably never been an exhibitor yourself, let me just say that the mere suggestion of changing anything on a demo machine—especially within hours of opening one of the biggest shows of the year—is enough to cause even the most stalwart marketeer to blanch and quiver. And to propose changing the operating system on their carefully-constructed

---

[*] IBM was finally launching the second version of OS/2 at this same show. Though IBM promoted it at their Comdex booth as a major competitor to Windows 3.1, the only visible marketing outside the show was a single OS/2 banner hanging from the IBM building in downtown Chicago, which was, ironically, directly across from our hotel.

computers is like asking a champion golfer to use something other than his lucky putter when he is only one shot from getting a double-eagle on the 18th hole and clinching the all-time scoring record in the Masters. It just isn't done!

Yet bursting with an enthusiasm that probably made Steve Ballmer weep for joy, we asked anyway. Surprisingly, we got a respectable number of takers. And not so surprisingly, a few of our evangelists got so caught up in the scene that they didn't even ask: to them, an unmanned booth was an open invitation to go right ahead and install the new system!

I have to admit that while I did install Windows 3.1 (with permission) for a couple of vendors, my heart really wasn't into this sort of thing. I was attending Windows World to give a talk on the OLE technology and to provide strategic counsel for various ISVs in attendance. With my role being more that of teacher and guide than politician, I was interested in truth, not promises. I simply couldn't afford the kind of hubris that is so common to marketing.*

After Windows World my work with OLE was more or less complete: with the help of my guidebooks and sample programs many ISVs were incorporating it into their products with little difficulty. Free to then turn my attentions elsewhere, I spent the spring and summer of 1992 applying the same process I had used with OLE to clarify other new Windows 3.1 technologies, like TrueType fonts and certain networking features. I authored more recipe booklets for these, wrote a good number of sample programs, and turned out a few magazine articles. I was also invited (or sent) to give presentations on these topics

* At least I came away with a good duffel bag that served me well for many years of travel. It fit perfectly in the overhead compartments of a wide variety of airplanes and, being hideously ugly, was both easy to spot on a baggage carousel and too embarrassing for anyone to steal.

at conferences and other programmer gatherings and made a few personal visits to individual ISVs.

Now there are three general rules for doing the kind of technical marketing I was involved with here, rules that apply both to product and presenter alike. One: shamelessly flaunt your best strengths. That much is obvious. Two: primp up the mediocre to make it all look special—how else do you get people really excited about esoteric software technology? And three—perhaps the most important where your employer's reputation is concerned: try to make even the most glaring faults look, at worst, mediocre. And if you can actually make those faults look like *features*, well, all the better!

I got good training in this from watching DRG's top evangelists at work, and I certainly could have imitated them. But playing spin doctor just isn't part of my nature—I didn't believe that everything coming out of Microsoft was somehow sacred. Yes, this attitude bordered on heresy, but I just couldn't help being openly honest and forthright about our technologies—especially when I was standing face-to-face with hundreds of other engineers! After all, it was that very honesty that made my work effective: when I spoke in public, then, I simply had to tell it like it was, for better or for worse.

I initially expected the worse. I wasn't following the unwritten corporate rules for one in my position and had every reason to expect reprimand or reprisal from the upper ranks of Developer Relations. These fears, however, never materialized; though I was ready for discipline, none came. Instead, my unique emphasis actually earned praise from my managers when it began to bear certain fruits.

For one, my "no-BS" approach won me a great deal of respect among the programming community at large. It was a good thing, too: I was young, and looked even younger. How young, you ask? Well, during my co-op tenure, when I was all of

nineteen years old, a caller with a deep southern drawl put it this way: "Now y'all listen t' me son…I've bin a-programmin' longer'n y'all bin alive!" Now, a mere four years later, his statement remained true for himself and probably a two-thirds majority of every audience I addressed. Many of the other third had probably been a-programmin' since well before my high school graduation. So for me to stand up there and tell these seasoned veterans how to do their work? I could have been crucified for presumption!

But my candor won their appreciation—so much so that after about eighteen months I was no longer looked upon as a cookie-cutter Microsoft Person whose only redeeming quality was his or her access to inside information.* It was now rather my own reputation that drew audiences: people came to my lectures and read my articles because they trusted me to help them understand Microsoft's increasingly complex systems and not just feed them some party line. In fact, throughout the latter part of my public-speaking career, my association with Microsoft was quite secondary; often it was altogether forgotten! I simply didn't act like certain standard-issue corporate lackeys that ISVs had learned to despise. I instead found my audiences open and receptive to what I had to share, and I like to believe that this had a meaningful impact on the success of our technologies.

A personal benefit that came from all this was increasing clarity. By being completely honest and by constantly looking at Microsoft's systems from the perspective of outside programmers, I could easily see ways to improve both our products and my own presentations. It's well known that the very act of trying to educate others helps one learn a subject better than

* Barring a singular exception that I'll relate in Chapter Twelve.

any amount of book study. Simply said, one's understanding needs testing. It needs some kind of exposure to the cold light of day—or the scrutiny of three-hundred nit-picky and scathingly practical engineers! Without this it's difficult to see your deficiencies, let alone correct them.

Now here's an interesting thing: for myself, most of those corrections came in the very moment I noticed a problem. The instant I realized a flaw in my presentation—or my comprehension—new insights and ideas immediately came to mind. It was as if they were just there, waiting for an appropriate opening in my consciousness. Sometimes in a lecture I would stop myself mid-sentence and ask everyone to wait a minute or two while I changed my PowerPoint slides to show a better (or more correct) way of doing things. On several occasions I even modified and recompiled my demo program on the spot.

At first I simply attributed this sort of thing to having a relatively decent lump of brain-meat floating around inside my head. After all, the mind comes up with instant solutions to new problems all the time, and the very attempt to explain something can be, as I said, very clarifying. But then I began to experience situations that not only challenged this particular rationalization but also challenged my very undersanding of intelligence.

The catalyst was the second version of OLE, introduced by Microsoft in its preliminary form in the fall of 1992. Whereas the purpose of version 1 was quite limited, version 2 was vastly comprehensive: the OLE 2 specification first expanded on OLE 1, then went on to transcend it altogether.

And when I say transcend I mean yes—in size, weight, and incomprehensibility! The new 300-page specification was overwhelmingly complex. Almost no one had a clue as to what the whole thing was really trying to achieve. It's not that the spec didn't contain all the necessary details, mind you—the problem

was that it did to a fault! In the very effort to articulate every minutia, the two software architects authoring this behemoth simply had no time to explain the "big picture" of OLE 2, a vision that they alone understood. As a result, ISVs were once again lodging their complaints with Developer Relations whose evangelists had sallied forth, once again, to spread The Gospel. And naturally, once again, it fell on my shoulders to make some sense of matters. So I dove right in and did my best to swim.

In reading the spec, I was led to believe that OLE 2 was a direct extension of OLE 1, offering the same basic features with a few added elements. So I took my OLE 1 programs and, over the course of the next month, labored to bring them up to the new standard.

Yikes! What a hideous chore! Yes, I got my programs to work…sort of. Pieced together with the electronic equivalents of bubble gum and lunchbox pudding cans, they more or less did what they were supposed to do. I kinda understood why I had to do certain things in my code, but just to implement the basic OLE 1 features with OLE 2 seemed twenty times more difficult! And there were large blocks of code that I'd just copied wholesale from the OLE team's samples because nothing worked otherwise. I had no idea what the code actually did and less of an idea why I needed it.

By no means, then, did I feel ready at the end of that first month to consider educating people about this monstrosity, let alone make it all easier for them. Given the choice, I would have liked to sequester myself for at least another trimester of study. But some higher-ups had decided that I should spend a week in the Canadian capital of Ottawa to personally help the programmers at Corel Corp. add OLE 2 features to *CorelDRAW* and a few of their other popular products. This was part of an effort to counter the false accusation that Microsoft's Systems Division (to which DRG belonged) gave special insider infor-

mation to our own Applications Division that allowed their programs to outperform the competition's. In reality, DRG gave much more advance information and personal assistance to companies like Corel than we ever gave to our own applications teams! Yet we needed definite and visible proof.

So off I went to Ottawa to give daily seminars on the essentials of OLE 2. Though I hardly felt qualified for the job, I was admittedly more qualified than just about anyone else outside the OLE team. Still, there were certain parts of the technology that had me totally stumped: I knew for a *certainty* that I simply didn't understand them at all….

Then something very curious happened.

Imagine yourself for a moment standing on the edge of a great black void, where solid matter disintegrates into nothingness, not unlike that climactic scene with Harrison Ford in the movie *Indiana Jones and the Last Crusade*. Now imagine extending your left leg and slowly leaning in its direction. Just as you are about to fall into the abyss, a stepping stone, solid and firm, appears beneath your foot! Then you take another step and lo! another stone appears, then another, then another, until you finally cross over to the other side of the dark chasm.

Well, that's essentially what happened during my first presentation: just as I got to a point of having to admit to everyone that I really didn't know what the hell I was talking about, or where I was even going with the whole presentation, the next step I needed to take suddenly appeared in the forefront of my mind. It seemed right, so I talked through that step. Then, once again facing the void, the next step appeared. Following that, another came, and another, and another, and another, until about half an hour later I found myself having just completed a coherent and seemingly accurate treatment of the subject at hand. It was so uncanny that I just stood there for a few moments in disbelief—I had no idea where it had all come from.

Later in the day I had a chance to check out everything I'd said in an actual program. Every bit of it worked. Somehow I had managed to correctly teach a subject that had been the epitome of confusion only twelve hours earlier.

Now had this been an isolated incident one might write it off as an anomaly and say that I really *had* known my subject but just hadn't yet formulated that understanding in words. Or one might say I was just following the logical sequence inherent in the system's design. But the experience repeated itself. Every day, in every presentation, I found an understanding of OLE 2 that simply hadn't been there before: I somehow knew what I knew I didn't. It was as if in my very admission of ignorance, in my very humility, a flow of grace suddenly appeared to compensate for my deficiencies. Time and time again, another power—clearly not a product of my own intelligence—had lifted me in the very moment I was about to fall.

At the time, I was not yet sensitive enough to see what was really happening. In fact, I can't remember even once sitting down to think through this experience as anything out of the ordinary. Even if I had, I don't think I would've known what to think about it. But it left a permanent mark on my consciousness: deep down I began to realize that something else was at work, a nameless and formless consciousness much larger than myself. I couldn't say what it was, but I felt a certain sense of gratitude for the help and guidance that it had given me in moments of definite need.

Now while I returned home with a much better understanding of OLE 2 than I'd had the week before, my knowledge was not in any way complete. My presentations at Corel covered only a portion of OLE 2's totality. Many things remained unclear as I continued struggling with the technology for the next seven or eight weeks.

During that period I had to give a few more presentations

on a three-city OLE tour with a couple of DRG evangelists. So once again I found myself standing at a podium facing my own lack of understanding, as well as a large audience!

This time, however, that humble thought of "Gee, I really don't know this…" didn't enter my mind. I figured I could fly as high as I wanted by myself—that *I* knew enough now that *I* should be fully able to give a good presentation under my *own* power, even if I had to blast my way through it. Well, it didn't work. I could tell from the complete lack of intelligent questions that my audiences were hopelessly confused. Perhaps it was compounded by the fact that I gave half of my talk before lunch (when the morning's coffee and doughnuts were ebbing fast) and gave the concluding half immediately afterwards. The lunch, a typically heavy affair, and the sheer drudgery of my thoroughly uninspired presentation proved themselves an admirable cure for insomnia. One time I almost slipped into a nap myself while standing at the podium as my digestive tract labored to break down large quantities of cheese! (Thereafter I limited myself to only the lightest of lunches—mild starvation tends to keep one awake.)

In any case, the flow of unexpected grace that I had experienced in Ottawa was not there. Why? It's very simple: it had been blocked by pride, by the thought that *I* could do it all on my own. Grace, you see, has to be invited and received: invited by the recognition and faith that while we might not know the answers ourselves, there is a Power that does; and received by the openness to allow that Power to work through us, according to its *own* will.

As a result, I was so faced with the stark reality of my own impotence that when this grace returned, I would be left with no doubt as to its true source….

CHAPTER TEN

Flash Flood

"What do we need to make OLE 2 successful?" On a typically dreary northwest Friday afternoon in January 1993, Jon Lazarus, Vice-President of Systems Marketing at Microsoft, was spelling out the problem: ISVs were struggling desperately with OLE 2; they had no idea what they were really supposed to do with the technology. And for that matter, neither did we. Our evangelical efforts in Developer Relations had been somewhat confused; we only succeeded in passing our own confusion on to everyone else.

It was increasing clear that if things kept going the same way they were, many ISVs would soon abandon OLE 2 altogether. This was simply unacceptable. So Jon called together everyone who had a key role in promoting the technology and asked for solutions.

"What do we need to make OLE 2 successful?" Jon's invitation opened the floodgates. "What we need are good, clear, technical papers," someone said. "And articles…" injected another, "that's what we need!" "And standard demo programs!" "More focused sample applications!" "Technical presentations!" "A developer's conference!" "And a book! We need someone to write a real how-to-do-it book!"

Whew! What a list! We definitely needed a lot…and, of course, we needed it all *now!*

I went home that weekend thinking about what I could do personally. I was certainly in a position to write articles, papers, and samples; it was already my responsibility to do so. I figured I could also help slog out a few more presentations and whip up a demo or two. Still, I had to wonder just how much I could really help the situation in the short term. For nearly five months now I'd been beating my head against a wall trying to grok this stuff with only marginal success. Writing a book was strictly out of the question…at the rate I was going, it would take me years to understand it enough to even write an outline.

With all the time I'd struggled with it, I was at least clear on one point: there was much more to OLE 2 than we were seeing. What we really lacked was a clear, high-level understanding that could be expanded on in the precise details. We needed to piece together every one of OLE 2's seemingly disparate elements into one coherent picture. But how? No one but the two software architects who had designed the technology would even know where to start, and they were so immersed in finishing OLE 2's monstrous specification that they didn't have time to even pretend to think about another project.

All these thoughts kept spinning around in my mind as the weekend progressed. "How do all these complex pieces fit together?" I asked myself. "How can it be simplified? What are the connections? There must be *some* way to make sense of all this!" More and more I focused my attention on these fundamental questions, seeking answers almost desperately. By Sunday, I was thinking of little else.

My wife Kristi probably noticed the strain of these mental gymnastics on my furled brow. With what I imagine was the most compassionate patience, she encouraged me to relax for a few hours while we visited her parents that afternoon. Realiz-

ing that my analytical mind was pushing itself to exhaustion trying to forcibly invent solutions, I was more than happy to oblige. Perhaps if I took a little break now I could find a new approach to my questions when I returned to work on Monday.

Aaaaaah. After enjoying a fine lunch and visiting with everyone for a bit, I felt much better. I had been able to put OLE 2 out of my mind for a while and, after reading the Sunday comics at the kitchen table, I sat back in my chair with a soothing sigh and found my mind at rest. Not a single thought disturbed the spotless sky of peace.

Then suddenly, as if magnetically drawn by my intense concentration upon it, the singluar question that I so deeply yearned to answer returned: *"How do all these pieces of OLE 2 fit together?"*

BOOM! A bolt of lightning flashed into my consciousness with tremendous power and in one timeless moment *I simply understood OLE 2*. Not just bits and pieces—*everything!* It was magnificent. Full-blown insights appeared in my mind with an indescribable thrill! Every knot untied itself! Every piece of the puzzle took its proper place! In an instant I knew the answer to my every question, even in those areas that I had yet to really study. And the entire architecture of OLE 2 stood before me with perfect clarity: an exquisitely simple foundation giving rise to mighty pillars that in turn supported the most elegant spires, each reaching into lofty technological heights.

It was astoundingly beautiful. So wonderfully exhilarating! And so *absolutely right*.

I was in awe: never before had such magnificent superconscious awareness coursed through my brain; never before had my entire being thrilled with such joy! It literally stunned me into stillness: I wanted nothing more than to just sit there

and absorb the profound bliss of these sudden realizations.*

Yet in that very same moment I knew that this inspiration was not for me alone: this vibrant consciousness had been *given* to me to help others and would have to be communicated to them in some material form.†

BOOM! Another flash! With immediate and utter clarity there crystallized in my mind the full-blown structure of a book that would present OLE 2 in its wondrous entirety. Each chapter would have a distinct focus on a particular piece of the whole; the sequence of those chapters would gracefully erect a complete architectural understanding, step by gentle step; and the sample programs accompanying each chapter would demonstrate each part both by itself and in the context of a larger application. It was the perfect way to do it! And as individual chapters could be easily shortened into articles—and as the whole sequence of the book provided the exact framework for a series of technical presentations—a focused effort to write the book would satisfy not just a few, but almost every piece we needed to make OLE 2 successful. Better still, I could start doing it all immediately!

In fact, I suddenly realized that I was the only person who *could* do it. No one else had the right combination of time, experience, and writing skills. Those who had the experience just didn't have the time. Other writers didn't have the experience.

---

* Johannes Brahms clearly describes similar experiences in *Talks with Great Composers* by Arthur M. Abell, which I only read years later. As he said, "I immediately feel vibrations that thrill my whole being…these are the Spirit illuminating the soul power within, and in this exalted state, I see clearly what is obscure in my ordinary moods."

† From Brahms again: "Those vibrations assume the forms of distinct mental images, after I have formulated my desire and resolve in regard to what I want—namely, to be inspired so that I can compose something that will uplift and benefit humanity—something of permanent value."

And my time was already committed to doing this kind of work anyway.

Clearly, this book was going to happen, and I was to be the instrument of its creation….

Like I said, this experience of "OLE Nirvana" (as Eric Maffei, former Editor-in-Chief of Microsoft Systems Journal later called it) happened in a flash as I sat at my in-law's kitchen table. As soon as I brought myself back to outer awareness (perhaps a brief minute later), I walked into the dining room where I'd left my notebook computer. I'd brought it with me "just in case" I had any good ideas during our visit.

I was again amazed. From the moment I sat down at the computer my fingers literally tap-danced on the keyboard. With an energy that was intense yet perfectly calm, the book's outline essentially wrote itself, line by line—the only effort needed on my part was to accept what was happening and just keep my fingers moving! In fact, I don't remember a single moment when I had to stop and think about what came next. Whatever was pouring through me simply had its own intelligence, no less astounding to me than the inspiration itself.*

Sometime during the three non-stop hours that I sat there "writing" this outline, my sister-in-law ventured by. Seeing how engrossed I was in what we normally consider "work," she asked, "Don't you ever stop and just have some fun?" What could I say? How could I explain to her that I'd never before experienced such utter delight? No mere pastime could compare.

---

* As the inventor Nikola Tesla said: "It was a mental state of happiness as about complete as I have ever known in life…ideas came in an uninterrupted stream, and the only difficulty I had was to hold them fast." (From *Tesla: Man Out of Time* by Margaret Cheney.)

The next morning I went to my office at 7am (I lived only a mile away) and began writing both the text for the book and the accompanying sample programs. To my deepening amazement, that joyful, effortless flow that was there with the outline continued, only now taking the form of paragraphs and source code. And it went on!—not just for a few days but day after day and week after week for a total of seven months. Without strain or exhaustion of any kind I wrote all morning long and did my programming throughout the afternoon. Sometimes the flow was so strong that I wouldn't have a moment's pause for up to eight hours at a time—the words just kept a-comin' and I just kept my little fingers a-movin'!

This, I discovered, was the real trick. Superconscious or divine inspiration depends upon willing cooperation with its own inherent course; self-assertion, or the desire to control its direction, simply stops the flow. I felt this clearly. Whenever I thought to impose my own will or opinion on matters, or tried to solve problems with my intellect alone, I got bogged down and felt the energy dissipating. When I once again gave myself into the flow, the whole process instantly became effortless and energizing: problems just seemed to solve themselves! (As St. Paul said, "God is not mocked.")

Such joy! Such delight! I was having more fun writing this book than I thought possible! "So why not make the book fun as well?" I thought. Yes, why not? So many programming books I knew started dry, proceeded dry, and ended dry. Not this one! Pushing the boundaries of style and convention (which my editors at Microsoft Press were gracious enough to accommodate) I pulled colorful quotations from playwrights, philosophers, *Monty Python and the Holy Grail*, Indiana Jones, and a handful of decently esoteric books on completely offbeat topics. Chapters began with themes on Tupperware, Cookie Monster, fishing, Darwinian evolution, and Howard Carter's excavation

of King Tut's tomb, each of which somehow managed to bring fresh insights to the discussion at hand. And rounding out the mix were a few world maps, a musical intermission (based on "Old MacDonald had a Farm"), a few satirical advertisements, a fair assortment of entertaining stories, and some outrageously bad puns. (Who says God doesn't have a sense of humor?)

By early September 1993, the final galleys of *Inside OLE 2*, as it was appropriately christened, were complete. A wonderful thing had been accomplished—not *by* me, but *through* me. I felt no egoic pride in the book's creation; I could only feel a wordless gratitude and delight for the experience of being immersed for so long in such boundlessly creative energy. Seven short months had witnessed the manifestation of nearly a thousand pages of print-ready text—edited three times over—and fifty sample programs. It was a rate of production well beyond what my poor editors at Microsoft Press had encountered before!

Yet unlike other authors I later met who came close to having nervous breakdowns after writing books half the size in twice the time, I didn't feel the least bit tired. I was rather energized and uplifted, feeling far better than ever before….

Now lest you get the impression that I worked relentless twenty hour days on this thing for seven months straight, let me make it clear that I "had a life" beyond the book. Quite a bit of life, in fact: my typical ten-hour workday, 7am to 5pm with lunch in the middle (usually), was actually below the Microsoft average and left plenty of time for other activities.

I took up walking as exercise, often cleaning up litter as I went. In the evenings I read about one book each week, many of which inspired new ideas for my writing. I was also actively designing and running an extensive Dungeons & Dragons campaign with an after-hours group at Microsoft.

I traveled a fair amount to speak at four or five conferences and visit a dozen ISVs. Kristi and I also took several vacations: five days in Illinois to celebrate my grandparents' 50th wedding anniversary, ten days touring Arizona, and a week in British Columbia's Queen Charlotte Islands where we attended a unique and deeply inspiring workshop with Dewitt Jones, the well-known nature photographer.

If this wasn't enough, Kristi had just earned her Master's Degree in Electrical Engineering from the University of Washington and had started a full-time job some distance from our apartment near Microsoft. Her commute meant I was honored with the privilege of cooking dinner every night, a task complicated by the fact that we'd recently become vegetarians.

What's more, our combined income now enabled us to buy a house as we'd long planned. We walked through a fair number of possibilities before finally settling on one that was still under construction. This meant running hither and yon to select carpets, fixtures, wood stains, and paints.

Coincidentally, we moved into our new home on the very day I finished reviewing the final proofs of *Inside OLE 2*. Was I then able to "relax"? Not at all: I finished those edits *en route* to conferences in Europe; I actually returned the moving truck on my way to the airport. Did I relax when I got home? Nyaah. I continued to read. I bought a grand piano and began to play an hour or two a day. And I got to learn about all the demands of a new house with two-dozen windows, oceans of carpet, an apartment's worth of hardwood floors, and a fifth-acre of lawn.

Then again, I didn't feel the need to "relax" with some kind of passive activity. I was reaping such a bounty of joy that it simply overflowed into everything I did!

Now you might have noticed that everything I've described so far took place *before* the book was actually published. By that time I can honestly say that I really did not care whether the book ever sold a single copy. I had already felt such tremendous blessings in the process of writing it that anything else was just a bonus.

Yet bonuses there were in abundance, as the blessings were only just now starting to reach out to others.

Inside OLE 2 hit the shelves near the end of November. The release was perfectly timed with a huge Microsoft Developer's Conference in Anaheim, California, at which OLE 2 was one of two major themes. As a result, the book literally became an overnight phenomenon. It hit the shelves and jumped right off of them!

Most computer books were considered successful if they sold 3000 copies over two years; at the conference alone we sold that many in the first two hours. In fact, it took only three weeks to sell out the entire first printing of 17,500 copies. Over the next eighteen months and two subsequent printings, *Inside OLE 2* sold a staggering total (for an esoteric programming book, at least) of 35,000.

Needless to say, the book accomplished every objective that we'd talked about in that January meeting. It helped make the conference—and the technology itself—a smashing success, not just in North America and Europe but even globally; *Inside OLE 2* found its way to the far reaches of the earth including Israel, Russia, Argentina, India, and South Africa with translations in German, Japanese, Chinese, and Korean.

As a result, I suddenly found myself with more friends that I knew what to do with and received a constant stream of email thanking me in some way for the book. Some of them told me it read "like literature." Many told of how much they'd laughed while reading it. I was delighted to know that even this highly

technical programming book was infused with the joy I felt while writing. That it touched others in this way meant more to me than professional satisfaction, more than fame, and more than money.

Not to say that these weren't forthcoming. It *was* deeply satisfying to have helped so many people with the book, and I had plenty of fame coming my way (and will be coming yours in the next chapter). As for money? Well, since I wrote the book on company time and had received my usual salary in the process, the book was "work for hire": Microsoft owned the copyright and I wasn't eligible for royalties from Microsoft Press. Given the success of the book, people pitied me for this "unfortunate" arrangement. But since I hadn't done it for money to begin with, it really didn't bother me in the least.

Some months later Jon Lazarus called me into his office. He'd originally opposed my writing the book, quite adamantly at times, concerned with the precedent that it might set within Developer Relations. Even for the first few months of its production he hadn't really given *Inside OLE 2* his support. But now he flat-out told me that he'd been wrong and that I had done the right thing. Thanking me for my dogged persistence, he handed me a special grant of stock options that in time became more valuable than any royalties would have been….

Joy, beauty, clarity, inner peace—"bonuses" of every kind—such are the natural graces of superconscious inspiration, of an open connection between ourselves and a higher consciousness along with the complete willingness to cooperate with its guidance. It revitalizes, it heals, and it just makes everything work right. Better still, such inspiration isn't the exclusive domain of a few elect souls: if it can come to an atheistically-inclined 24-year-old programmer in the middle of Microsoft's technological

marketing efforts, it is truly available to all who express the energy, concentration, and love necessary to attract it and who are open and courageous enough to receive it.*

Yes, *courageous*. Drawing on creative inspiration and learning to make it a natural habit means taking the responsibility to manifest that inspiration in a tangible way. This implies a great deal of effort and self-sacrifice along with the willingness to expose yourself to public scrutiny. More importantly, though, it also means the willingness to expose yourself to how *you* might be changed in the process!

When the whole *Inside OLE 2* project began, you see, I had little visible interest in spiritual matters. Political and social issues occupied my extracurricular thinking and religion, in my mind, was a negative, repressive influence. I'd just read several books about the Crusades in which religion, as "practiced" in that context, seemed little more than a widespread excuse for hatred, prejudice, and violence. As for God, I saw "it" in terms of a blind socio-psycho-political force that had been used, almost exclusively, as the justification for a dizzying array of nefarious deeds. God and religion, in other words, were barriers that I sought to overcome as much as those I faced with the OLE technology.

In the process of writing the book, however, I couldn't deny that there was this tremendous power—a higher consciousness—flowing through me. It couldn't be dismissed, nor could it be buried under some clever philosophical definition. And while it came from within me, *it was clearly not my own.*

My skeptical belief system had no provision for this sort of thing—yet to suppress that flow or shut it out would have

---

* For a more complete discussion of this subject see *Art as a Hidden Message* by J. Donald Walters.

meant not writing the book. It would have meant leaving thousands of desperate programmers out in the cold. It would have meant disconnecting myself from a source of energy that was giving my life purpose and joy.

I couldn't think to entertain such alternatives for even a moment. My only choice was to just let go my opinions and let the flow carry me wherever it would.

Throughout the period I've described in this chapter, my conscious attention was so much occupied by what was happening outwardly that I barely noticed what was happening inwardly: new kinds of thoughts were appearing in my mind, and I was becoming more and more open to new realities. Many of these ideas I found fascinating and meaningful, and I would have loved to explore them in depth. As it was I ould only jot them down in my personal journal—I had to keep my focus on writing *Inside OLE 2*.

It was only while writing this present story that I really looked at those notes again. What they reveal is that God wasn't just giving me a book, he was directly transforming my consciousness. I began to ponder the questions of who I was and what life was really all about. I questioned the complicated modern lifestyle and its advertisements that equated quality of life with the shoes I wore, the news I read, and the beverages I drank. And I questioned many other popular values, especially those related to justice, morality, and "right living." In short, my inner search for truth became significantly more intense.

I also began to read more spiritually oriented books at this time (including various scriptures) and, amazingly enough, to think a great deal more about God. For years I'd been trying to write him off as a social fabrication, but he just wouldn't go away. Yet I still couldn't think of him as being somewhere "up there," forever distant and unapproachable as he's so often presented. More and more I was searching for some concept that

was present, tangible, and intimate.

My attitude toward God and all other spiritual stuff thus shifted during this period from outright rejection to at least a probationary acceptance. As I continued to be immersed in the deeply spiritual (though not "religious") experience of writing the book, I came to understand that spirituality was not a matter of belief or ritualistic worship—it was rather demonstrated by one's day to day choices and experience. As Jesus said, "Why do you call me, 'Lord, Lord' and do not what I tell you?" My actions were what mattered, not some intricately chiseled theology. My thoughts mattered too—was it not sheer hypocrisy to talk about kindness while clinging to unkind attitudes? One must do more than *preach* peace, he must first *become* peaceful in himself!

More and more I was coming to realize that spirituality is a wholly inner process, not an institutional one. The answers I wanted to life simply had to be *inside* myself, just like the inspiration—and the very words!—for *Inside OLE 2*. As Jesus also said, "Neither shall they say lo here! or lo there! but behold, the Kingdom of God is within you." But it wasn't that *I,* this little ego, was the source of Truth. My little body simply could not contain anything worthy of the name "Kingdom of God." But I was beginning to understand that somewhere within my being—and within every other living creature—I had a connection to an overarching unity…a unity that, for lack of any other name, I *could* call "God"!

Inside OLE 2, then, was the channel through which God really began to throw open the windows of my receptivity and awareness of a greater reality. Through the torrent of inspiration that produced the book, the Divine Gardener had washed away many weeds that were choking my soul. And in the now fertile soil of my open heart, where I noticed a slight though not wholly physical sensation, he was planting the seeds of his

presence. Those seeds would still need a few years to sprout, and during that time my general thinking would revert to politics, social justice, and other such concerns.

But sprout they would, changing my life in even more complete and wonderful ways.

CHAPTER ELEVEN

Name, Fame, and Guru Game

"What is the best part about writing a book?" In the wake of *Inside OLE 2's* stunning success I was frequently asked this question.

"Fame," I replied.

"And what is the worst part about writing a book?"

"Fame!"

In Microsoft's Developer Relations Group, we often referred to ourselves—jokingly, but with some truth—as Microsoft Lackeys. When we spoke at conferences, visited other companies, and interacted with the press, it was not as individuals but rather as a breed of interchangeable victrolas for The Microsoft Gospel. It didn't matter who you were, personally, so long as you fit the right mold, behaved according to expectations, and wore—literally at times!—the same standard-issue logo shirt.

Public exposure in this capacity was part and parcel of almost every position within DRG. It certainly was true for me through the first eighteen months of my tenure there. Being of the cookie-cutter variety of young-and-reasonably-intelligent Microsoft representatives, my early invitations to speak at conferences came not through any special merit of my own but only because the organizers wanted A Microsoft Person. And since it was DRG who generally chose an appropriate Lackey,

it occasionally fell upon my shoulders to play the role.*

People thus came to these early talks of mine only because of what I symbolized: an understanding of our technologies from the inside-out. People knew that spending an hour or so listening to my techno-babble could actually spell the difference between failure and success. And they knew that a few minutes of my personal attention could help solve a baffling problem that might otherwise demand days or weeks of grueling anguish. Beyond that, however, everything else about me—name, personality, hair color, whatever—was quite irrelevant and not particularly special (except, perhaps, for my head of red hair). I was treated like Any Other Microsoft Person and people were just as likely to give me a wide berth as they were to peg me with a question. Certainly no one ever bothered to ask for an autograph.

Being somewhat shy and introverted by nature I was happy to draw attention to Microsoft's technologies and not to myself. I wanted to help people understand and use those technologies to their fullest capacities. As my manager Friedrich had said, "The purpose of your job is to make yourself obsolete." I figured that the more I helped others develop their understanding—and even exceed my own—the less they would depend on me, in particular, for anything. I could then retire to my comfortable little office in Redmond and pursue other projects without pain or inconvenience to anyone.

---

* Sometimes this process had amusing side effects. One day DRG's director received an email from a group in Honolulu, Hawaii, requesting that someone come out for a week—all expenses paid—and give talks on Windows NT. He forwarded the request to us with the rather unnecessary question, "Anyone want to go?" In instances like this the determining factor was not willingness or competence but reaction time: at least a dozen eager volunteers stepped forward within two minutes—most of whom knew little or nothing about Windows NT but were fully prepared to learn!

With Friedrich's guidance I sought to avoid the common pitfalls of many "experts" who pride themselves on their experience, who accept and even desire undeserved adulation, and who protect their position as the "ideal" toward which others should aspire. We knew that such attitudes would only lead, in the end, to stagnation and—indeed!—true obsolescence. So I sought instead to place myself in a position of service to others, sharing with them everything I learned and developed. In this way I had nothing left to protect and would find myself surrounded by many appreciative friends rather than a "following" that I'd be ever fearful of losing.

It was a fairly straightforward process. I first spent a few weeks or months developing an expertise with some particular technology. Then I only needed to write a paper or two, give a few talks, publish an article and viola! it seemed that people caught on and took care of themselves. And while I sometimes gained a bit of public recognition along the way as an expert in these things, it was only temporary. People soon forgot about me and got on with the task of creating interesting and innovative software.

Then came my work with OLE version 2, a much more extensive project, to say the least. In DRG we promoted this technology far more than most others so it was attracting a great deal of attention. People clamored to learn everything they could and I was there to provide it. This time, however, there was a new dynamic: throughout most of 1993 I was, in all honesty, the only public person who really understood OLE as a whole, as the two architects designing the thing were safely hidden in their comfortable little offices in Redmond. So if you wanted to learn about OLE, you came to my talks and read my writings: there simply wasn't anything else available save the heady tome of the design spec. Indeed, the draft chapters of *Inside OLE 2*, which Microsoft Press magnanimously allowed

me to disseminate prior to publication, were simply *the* source of digestible information on the new technology.

In the utter absence of alternatives, then, my name and image soon became a virtual synonym for OLE itself. I was, in the public eye, the "OLE guru." While I didn't care about such a title, I was grateful that this notoriety gave me all the more opportunity to share my understanding with increasingly large audiences. I was equally pleased that as 1993 wore on, people who came to my talks really seemed to "get it." Never once did I feel that they were paying all that much attention to me, personally, even as the OLE figurehead.

Never, that is, until the big Microsoft Professional Developer's Conference in mid-December. This event, which coincided with the publication of *Inside OLE 2*, was held at the cavernous Anaheim Convention Center in southern California—directly across the street from Disneyland. It was the largest conference that DRG had ever put on and probably the most intense. Our central theme at this veritable circus was simple, clear, and direct: "Win32 and OLE 2! Win32 and OLE 2!"* With this slogan we vivaciously promoted Windows NT and OLE, hyping the crowd of over 8,000 engineers, analysts, and managers to heights heretofore unrealized. And as the OLE guru—as the author of what had already become the essential holy writ of the technology—I was asked (along with another Microsoft trainer, Cathy Linn) to give an all-day pre-conference tutorial to explain the details.

---

* Win32 referred to the internal 32-bit architecture of Windows NT (and then Windows 95, 98, XP, Vista, etc.) which offered much better performance than previous 16-bit architectures (e.g. Windows 3.1). A 32-bit architecture could manipulate data twice as fast as before but software had to be specifically rebuilt for that architecture to realize the benefits, hence our big evangelistic push at this conference.

Now while I was fully accustomed to giving presentations by this time, having long since shed any lingering remnants of stage fright, I still wasn't quite prepared for this one. As I took my place at the podium, each of the fifty some-odd spotlights that illuminated the stage seemed intent on being hotter than a Saharan sun. Massive thirty-foot projector screens magnified my talking head to proportions literally, way literally, larger-than-life. And beyond the first few rows of people, which were all my acutely contracted pupils could resolve, a dark, heaving mass of techno-geeks, numbering close to six thousand, filled every square foot of floor space in the hall and considerable areas of the adjoining passageways.

Of course I was delighted with this whole opportunity—I had spent the better part of a year trying to impart my understanding through the pages of my book. Through my talks I strove also to pass on the inspiration and joy I had experienced while writing it. To now have such an enormous audience for this purpose was simply magical. It gave me the energy and enthusiasm to present the finest in-depth talk I ever gave on the subject.

This aspect of fame—the ability to share one's joy with so many receptive souls—was indeed the best part about writing a book. I was in a position to give away absolutely everything I knew to everyone that mattered. And once I did so I figured they could all carry on without me. I could then withdraw from the public eye and retire, again, to my comfortable little office in Redmond.

Minutes into my talk, however, I discovered that my fanciful dreams of obscurity were utterly naïve. There were these three guys sitting in the front row. Each of them watched my every move with glazed eyes and the peculiar smile of infatuation. To my absolute horror I suddenly realized that these *same three men* had sat in the *same three seats* with the *exact*

same smiles at no fewer than three talks I had given during the past three months—in three different cities! And going on the assumption that they weren't attending this one to learn more about OLE, I was left with only one inescapably nauseating conclusion: I had groupies! Yuuugh.

It got worse. After my talk I was besieged by an unsought throng of admirers. All the hype, all the spotlights, all the big screens, the huge lecture hall, and my own enthusiasm succeeded in making me the star of the show. Sure, I had been famous before, but, you know, not *famous*-famous! And now anybody and everybody who cared about OLE knew who I was, knew what I looked like, and knew where to corner me.

I was hard to miss: besides having a distinguishing head of red hair I was also wearing this fabulously ugly standard-issue red shirt. It was once Microsoft's practice at developer conferences to identify the "technical" folk—that is, those who were suitable targets for the unceasing barrage of questions—with flaming red shirts. I always thought it had something to do with the old 1960's episodes of Star Trek in which the expendable security officers, the ones who were always getting shot if *anyone* was getting shot, wore red shirts. In any case, I despised my GI apparel—for one, I look awful in bright red, and two, it left me nowhere to hide. I couldn't go anywhere near the convention halls without getting mobbed; not just by those who simply had honest questions but by those who wanted to meet me, touch me, breathe the same air, or in some way come into contact with my guru-aura of fame and glory.

It wouldn't have bothered me quite so much if people hadn't somehow lost the ability to relate to me as a human being. Good God! All I did was write a book and give a decent lecture for a change. Nothing else about me was different! But now people weren't even willing to acknowledge, for example, that I too was human and that I too, on occasion, needed to use the

bathroom. At one point, after fighting my way to the entrance of a facility, I had to say "Look, you can come in and *watch* if you want, but I have *got* to go!" Others weren't willing to acknowlege that I actually slept at night—I was awoken one morning at 3am in my hotel room by an obnoxiously loud telephone. The man on the other end immediately demanded that I answer some obtuse technical question. "But it's three o'clock in the morning!" I protested. But he persisted. I finally decided that the quickest way to get rid of this clod was to just answer his blasted query. After that I learned to always disconnect the phone before going to bed.

In the face of "greatness," whether real or imaginary, most of us have a tendency to start acting really stupid. I say this from experience on both sides of the equation. The first (and only) time I personally met Bill Gates, for instance, I was a complete dolt. It was at a party for summer interns held at Bill's house back in 1989. When Bill came in I happened to be talking with this brilliant Ph.D. candidate from M.I.T. We were the first people that Bill came to greet and, as a young college undergraduate doing what was, relative to my companion, entirely inconsequential work, and standing now in front of the world's wealthiest man with a reputation as one of the greatest technical geniuses of the modern era, I was at an utter loss for words. I think I managed to croak out a "Hello" before he got into some profound theoretical discourse with the M.I.T. issue involving words that I couldn't have found in a dictionary were it embedded in my brain.

Buh…buh…buh…buh…holding Bill in such high esteem simply left me speechless. I didn't want to look like a fool, nor did I want to be a jerk. So paralysis set in and I just seized up. At least this was the decent thing to do. Had I been unable to maintain silence, I'm sure I would have opened my mouth only to stick my foot in it: presuming, for instance, to be his equal,

or worse yet, acting like I somehow *owned* him by trying to monopolize his attention. I'm sure that would've gone over with Bill about as well as deposition-hungry lawyers.

The other option (besides paralysis), and an even worse one in my mind, would have been to start gushing—that is, to curry favor through outright flattery. As it's been said, imitation may be the sincerest form of flattery, but flattery is the most useless form of praise. Somehow we get it in our minds to set people up on a pedestal and virtually worship them for their accomplishments, exclaiming over and over again how "wonderful" and "marvelous" they are compared with our untalented, uninteresting, and unimportant selves. The reason for this, I think, is very simple: gushing betrays the sad fact that many of us are actually threatened by greatness even if we're also, to some extent, inspired by it. Greatness of any kind in another human being—especially one who isn't all that different from us—is a reminder of what we could ourselves accomplish if we but invest the necessary time and energy in the appropriate direction. "Why haven't you, then?" That's the uncomfortable and even embarrassing challenge that we're not always ready and willing to meet. So what's the alternative? Put them high on pedestal: if something can be exalted beyond reach it ceases to threaten our complacency and self-satisfaction![*]

Unfortunately, many teachers, experts, and other celebrities are all too happy to go along with such nonsense and make themselves the focal point of people's devotion, however poorly

---

[*] "Do not your scriptures say," Jesus reminded the orthodox religionists of his time, "'ye are Gods'?" Or as he said in the Sermon on the Mount, "Be ye therefore perfect, even as your Father which is in heaven is perfect." Yet how many still persist not in trying to become Christlike themselves, as Saint Francis of Assisi exemplified, but go on and on merely singing praises *about* God instead of even singing *to* him!

placed. Playing the "guru game," they enjoy for a time their place in the limelight and in the hearts of their following. Sadly, though, this game has no winners. Destructive cults aside, I'm particularly referring to a less dramatic but more insidious effect: if we are unwilling to come up to or even exceed the level of someone we admire, we lose a precious opportunity to learn from them and grow. Similarly, we actually prevent *them* from growing and changing as well. From whom can they learn if no one is willing to excel further and set an even higher example? How can they branch out into new directions if we enslave them by our expectations?

Yes, this was the *worst* part about writing a book! Whereas I sought to lift people up, they wanted—even subconsciously, perhaps—to lift *me* instead, giving me far more credit than I actually deserved. Despite my best efforts to help people develop their own understanding, I found myself standing on the very pedestal I sought to destroy.

I don't know what would have happened if I had been left to face all this on my own; at some point I might have succumbed to the pressures of the game. But by God's grace I wasn't alone: in early 1992, when I first began making public appearances, I had somehow become instant friends with a couple of the other computer industry "gurus." The first was Richard Hale Shaw, widely recognized for his expertise in programming languages and development tools. The other man was the veritable patron saint of our profession, Charles Petzold, whose perennial best-seller *Programming Windows* (first published in 1987) was our Bible. For more than a decade, anyone who knew how to write a Windows program learned it from Charles. Indeed, the book remains in print to this day and is still going strong.

I was never exactly sure why they took me in as a friend. It was really quite extraordinary. Why did such revered veterans pay any attention to a 23-year-old Microsoft poster-boy like

myself? Why did they essentially accept me as their equal after I'd written all of one or two articles and given maybe three real lectures? And why did I continue to find myself in their company in the months and years ahead? I can only think that they could see where I was headed. They knew from personal experience that the apparently rare ability to both learn about technologies and explain them clearly to others would pull me into the public eye and into the spotlight of public expectations. It can only be from a deep sense of love and caring that Richard and Charles took it upon themselves to guide me through these treacherous waters and show me the proper attitudes for one in their position. Both of them demonstrated that lovingly humble spirit that is so very essential to teaching and sharing; with a genuine concern for the needs of others they offered themselves to the misunderstandings of popular fancy. They became experts that they might make more experts; they suffered that others might be spared the many miseries of our profession.

In their company I was able to develop these attitudes, avoid the popularity trap, and simply be myself.* Good thing, too, because for me, the Anaheim conference was only the beginning! The immediate success of *Inside OLE 2* and its continued success through all of 1994 managed to elevate my status to ever-new heights, beyond even the ostensibly serviceful role of "OLE guru." A few months later, for instance, during one of the many conference trips that had me away from home for a total of nineteen weeks that year, I was sitting in the

* This was especially true during the PDC when Microsoft rented the whole of Disneyland for our exclusive enjoyment one evening. Finding me alone, Richard pulled me into the little band of celebrity types that he'd gathered together as a kind of mutual defense strategy against the admiring crowds. In this group I also became dear friends with another author/guru, Bruce Eckel, whose companionship in the years ahead I treasure in my heart.

lobby of the San Jose Marriott having a pleasant conversation with Charles Petzold. We were actually talking about how nice it was that we, with whatever trifling fame we had acquired, weren't *so* famous that we could still walk out in the street without being accosted by paparazzi. Just then one of the conference attendees walked by and about fainted. "Ohmigod," he cried, "Wow! Seeing *both* of you together at once! The OLE god and the Windows god! Oh! Oh! Oh!" My only cause for cheer at that point was that I hadn't eaten recently…yuuugh!

Then there were the so-called "book" signings. Besides adding a scribble or two to plenty of my own, I was also asked to autograph other people's books. *And* the backs of business cards. *And* nametags, napkins, envelopes, random slips of paper, and one fellow's forearm.* A few people even wanted to take photographs with me! I hope to God those pictures didn't end up in some kind of shrine.

Then there were all the other titles that got attached to my name. Because there was (still) no other public figure associated with OLE, it wasn't enough to simply be someone who could explain the stuff: I was given credit for actually inventing it! Since the brilliant minds who had really created OLE remained safely anonymous, per Microsoft policy, in their comfortable little offices in Redmond,† I was left as the only possible target for a host of creative but wholly bogus honorifics. Several magazines referred to me as "the creator of OLE" or

---

* The latter turned into a running joke between Bruce Eckel and myself whenever we sat together for book signings. Knowing of the incident, Bruce loved to tease me by loudly announcing that "if you don't have a book, Kraig will sign your butt!"

† Microsoft typically hides their superstar programmers and software architects from the public lest they become harassed by the endless calls from headhunters and the other such annoyances that assaulted my own ears.

"one of the original developers"; to highlight an interview I gave, a European magazine described me on their cover as "The Mind Behind OLE"; and in an ad for my book, Microsoft Press even saw fit to crown me "one of the great programming minds" of the modern software industry.

Sigh. What could I do? I had little choice but to let people indulge their fantasies. To refuse praise, to refuse the courtesy of an autograph (even on a forearm), and to continually counter every undeserved label would be rude, hurtful, and obnoxious. I simply had to accept it all as part of the role. After all, people *were* being helped by my efforts and needed *some* way to express their gratitude, even if those expressions sometimes bordered on the ridiculous.

At the same time, I didn't have to accept it all for my own ego: I knew I was only a channel for something greater than myself, for the powerful inspiration that carried me through the production of *Inside OLE 2* and indeed through everything that followed in its wake. So when people came to me with praise I tried to inwardly pass their gratitude on to the source of that inspiration, though I did not as yet even have a name for it. To further emphasize the thought that people's praises *were* expressions of gratitude, I also trained myself to habitually say not "thank you" but "you're welcome" in return. And they *were* welcome: I was ever ready to give others as much as I possibly could for their continued benefit and growth.

In the end, it's important to understand that any teacher or guru worthy of the title is *never* interested in showing off themselves or their special talents. They are only interested in the upliftment of others. The very term, *guru*, in fact, comes from the Sanskrit word *gur* meaning "to raise or uplift." So forget about fame! Forget about glory! Forget about praise! If we would show them our gratitude—and truly relate to them as they would have us do—then we should offer our friendship,

our support, and our sincere willingness and effort to become as they are, to fulfill our inner potential as channels through which love and joy and understanding can be shared with a world that desperately needs it. This, above all, is what any true teacher, a true spiritual teacher especially, wants us to discover.

Eventually I did find that my job with OLE was complete and that I could retire—not just to my comfortable little office, but from Microsoft altogether (as told in Chapter Sixteen). Today I'm happy to see that the OLE technology has become so pervasive that it's simply a fundamental part of almost every new technology coming out of Microsoft. I'm also happy, believe it or not, to see that my name and writings don't even appear as references in more recent books and articles. I haven't seen any OLE-related email for quite some time now, and it's been many blissful years since I've had to sign an autograph. (I will admit, however, to signing whatever copies I come across in secondhand bookstores!)

And just when I was drafting this very chapter, my paternal uncle, a wedding photographer in Springfield, Illinois, was visiting Seattle. He told me about a man who had come to hire him for his upcoming wedding. To make sure that he could accept the job, my uncle told him about his travel plans.

"That's great!" the man replied. "Where are you going?"

"After visiting my brother in Seattle," said my uncle, "we'll be driving down the coast to San Francisco."

"How about that!" exclaimed the man, "I'm moving out to Seattle shortly after my wedding."

"Oh, how interesting. Did you get a new job?"

"Yes. I'm actually moving to a suburb called Redmond. I'll be working for Microsoft."

As soon as he mentioned Microsoft, the man fell silent for a moment, his mind jumping through a few associations with my uncle's patronymic.

"By the way…" he said, venturing a further inquiry, "you don't happen to be related to a 'Kraig' Brockschmidt, do you?"

"As a matter of fact, yes," my uncle acknowleged. "He's my nephew."

"Well you should tell him that I've read and studied everything he's ever written about OLE. It's thanks to him that I got this position!"

Yes, dear reader, I finally knew for certain that I did my job right. Whatever fame, name, and "guru" status that *Inside OLE 2* and its subsequent second edition brought me meant nothing. What mattered was that I was able to help someone take a meaningful step upwards in his career and in his life.

And *that* is the best part about writing a book!

CHAPTER TWELVE

Purpose

"But I say unto you, Love your enemies,
bless them that curse you, do good to them that hate you,
and pray for them which despitefully use you, and persecute you."

—Jesus

"In other news, the Evil Empire of Redmond today announced that…" Sigh. It was a rare week that passed without some kind of unrestrained verbal assault upon Microsoft in the industry tabloids. With its unprecedented success, high energy, and occasionally unorthodox methods, Microsoft tends to evoke emotional extremes. People love it or they hate it. Whether it has some correlation to one's profits from Microsoft stock, I do not know. But one thing is certain: Microsoft is no stranger to attack. It is a testimony to the company's internal strength and the quality of its leadership that it has continued to thrive in the face of so many challenges.

Prior to the advent of governmental anti-trust activity at the turn of the millennium, one of Microsoft's greatest challenges came in the early 1990's. Apple Computer, Inc., whose Macintosh line was losing market share to Windows, sought to prevent Microsoft from using similar user-interface features by

suing over copyright infringement. (The suit, filed in 1988, originally targeted Windows version 2, and was expanded when the highly successful Windows version 3 was released in 1990). As is typical in such cases, Apple asked the courts to halt the sale and distribution of Windows.

We were a little shocked when the news reached our ears. It had always been one of our principles at Microsoft to compete through innovation, not litigation. It really did sadden us to see a great company like Apple taking this approach because we knew it was in their power to innovate if they chose. It was also sad because Apple had often cooperated with us in the past; it was difficult to be forced into conflict.

It was a painful time. In the hostile environment created by the lawsuit, anyone who had bones to pick with Microsoft came out of the woodwork and let loose their criticism. Every week brought new insults to Microsoft from the weekly columnists whereas Apple was made out to be the force of righteousness. For those of us who represented the so-called Evil Empire in public, this certainly made for some interesting road trips!

Opinions aside, Microsoft had to fight and fight they did. Not only did the company successfully block the stop order but gradually whittled Apple's list of 189 claims down to 10, 5, 2, 1, and finally zero. When it was all over Apple had spent several years' worth of time and research capital for nothing. What's more, over the course of the lawsuit the personal computer market expanded tremendously, but Apple wasn't there to claim their rightful share. That share pretty much fell into Microsoft's lap: Windows emerged the victor by a factor of ten to one over the Macintosh, and Apple was sadly left with various financial struggles.

While I had to personally bear an occasional insult on behalf of my employer, I wasn't involved enough for it to affect me all that much. Indeed, Apple's lawsuit goaded many of us to

concentrate even more passionately on our work, be it creating better software or helping others understand it more deeply.

Other challenges, however, hit much closer to home and struck a stunning contrast to the accolades I received as an industry expert. In 1994, for example, another lawsuit arrived on behalf of Wang Corporation. This one was aimed specifically at OLE, now the centerpiece of my career. Wang was going through Chapter 11 reorganization and their creditors were doing their rightful duty to find anything of value in the corporate files. Discovering a few patents to which certain design elements of OLE bore a striking resemblance, Microsoft was promptly sued for infringement.

The columnists of the computer weeklies were, of course, delighted. They flooded their pages with new attacks on Microsoft and OLE and even used this latest lawsuit to bolster the outlandish but oft-reiterated idea that Microsoft had a megalomaniac desire to control the world. Someone, in fact, circulated a rather creative and fairly convincing "internal Microsoft document" that detailed our "plans" for a New World Order under which Bill Gates and his henchmen would issue directives from "Building Seven, a secret subterranean bunker hidden beneath Microsoft's Corporate Campus."* To make the story even better, the anonymous author of this document showed how certain architectural elements of OLE itself would be the means for the takeover!

Even as ridiculous as such claims were, the constant flood of negativity that followed in their wake started to hurt. For whatever truth there was in Wang's allegations, and for the definite non-truth in the rumors of sinister mind-control, OLE

---

* As I mentioned in Chapter Six, Building Seven had been designed but couldn't be built due to setback restrictions.

was my work. It was the only project I really cared about. In both the public eye and in my own, OLE was very much my identity. In a very real and tangible way, an attack on OLE was an attack on me and the life-blood I had given to it.

Fortunately the Wang suit was quietly resolved a couple of months later by a mutually beneficial and friendly agreement. Greatly relieved, I enjoyed the peaceful absence of animosity for a time. Then, not unexpectedly, came another, even more focused attack.

Enter a coalition of Microsoft's primary competitors of the time: IBM, Apple, and Novell. These three were apparently concerned (I can't speak to their real motives) that OLE, over which they had absolutely no influence, was becoming an industry standard. So under the banner of a non-profit corporate body called the Component Integration Laboratories (CI Labs), this triumvirate and two other relatively neutral companies set out to create a competing technology called OpenDoc. So far as its outside advocates had it, this would not be an evil "proprietary" technology like OLE: OpenDoc would rather be an "open standard," designed by a committee of the founders of CI Labs and anyone else who coughed up the necessary $50,000 for a voting seat on CI Lab's board. This was, OpenDoc's supporters seemed to imply, infinitely preferable to having a few deranged toadies within Bill's neo-fascist cult determine the industry's future!

Given that attempts to create industry standards by committee have usually been unmitigated failures, most of us in Developer Relations (and Microsoft as a whole) simply ignored OpenDoc.* On technical grounds alone, we pretty much knew it

---

* Excepting James Plamondon (see Chapter Nine) who spent a year directly addressing the OpenDoc challenge.

would never really fly.* Even so, there was no escape for us. OpenDoc's supporters in the industry (and press) often held up our silence as a clear indication of Microsoft's arrogant, non-cooperative nature. They set up OpenDoc as the new messiah that would save the industry from Der Führer Gates and his party hacks. They even managed to make "proprietary" a dirty word. And all of it showed up in the weeklies whose columnists seemed to revel in this fresh opportunity to renew their favorite pastime of reckless Microsoft bashing.

Personally speaking, I had to field a number of OpenDoc-related questions at various conferences and over email. Occasionally someone would openly insult OLE to my face and lavish praise on OpenDoc. They seemed to forget that OpenDoc didn't even have a final design—let alone a working implementation—and that its proponents could say and promise anything they pleased! In any case, it wasn't an easy thing to counter unsubstantiated claims…all I could really do was take the high road and just keep talking about the benefits of OLE–grateful, at least that the insults weren't really personal.

Not, that is, until I gave a talk in the spring of 1994 at a conference called Object World.† This conference was the main stomping ground for another industry movement centered on something called "Object Technology," the prevalent standards

---

* It was also curious that CI Labs seemed consciously organized to thwart Microsoft even if it joined the coalition. The group's by-laws required an 80% majority vote of its ten-member Board to ratify any proposal. This struck us as rather odd given that almost every other committee in the world operates on a two-thirds majority. Then we realized that this particular rule effectively rendered Microsoft impotent: even if Microsoft joined the Board itself and bribed or created six other companies to vote the Microsoft line, Novell, Apple, and IBM would form an opposing minority making an 80% pro-Microsoft majority impossible.

† OpenDoc never really happened. CI Labs officially shut down in July, 1997.

for which are contained in a specification for the "Common Object Request Broker Architecture" or CORBA, for short.

CORBA was something of a sacred cow to the Object Technology advocates, and for good reason: it remains one of the few successful standards created by a diverse consortium (known as the Object Management Group or OMG). Indeed, it seemed to become almost something of a religion among certain sectors of the industry, complete with passionately defended dogmas. So far as I could understand it, there were certain things that passed an unwritten acid test for TRUE OBJECT TECHNOLOGY and certain things that did not. OLE did not. While OLE was ostensibly concerned with software "objects,"* OLE's sort didn't adhere to the sanctioned forms of the orthodoxy. They were not TRUE OBJECTS; therefore OLE was not a TRUE RELIGION. Therefore some CORBA and Object Technology pundits went so far as to brand OLE a virtual HERESY and Microsoft an ENEMY OF THE CHURCH.

If you don't understand all this, don't worry—neither did I! To this day, I still don't know why people made such a stink about it. I only know that back then the Object Technology crowd seemed generally hostile to anything Microsoft did, perhaps for no other reason (from our point of view) than to slow down OLE from becoming a *de facto* industry standard.

Anyway, the battle over this TRUE OBJECT stuff had been heating up considerably in the first half of 1994 and Microsoft decided it was high time to meet the challenge with a large presence at Object World. (We had never even bothered to *attend* the show before, let alone present our story.) This meant that a certain someone got to give a talk about OLE….

---

* Loosely defined in the lower-case form as self-contained bundles of program code and data that behave according to certain well-defined characteristics.

Without realizing that I was a sacrificial lamb, I innocently went to the show, got myself situated in the lecture hall, and stood at the podium to deliver my usual introductory presentation, "What is OLE?"

Now its common during technical lectures for people to ask questions when they are confused or need clarification on some point. About halfway into my presentation a suited gentleman stood up and asked something about databases. I can't say any more than that because every other word in his question was, to me, indistinguishable from Swahili. So I followed standard procedure and asked him to restate his question, but I still didn't understand anything. At this point protocol demanded that I not waste everyone's time—I politely asked him to see me after the talk was over.

He then stood up, huffed and puffed, and stormed out of the room in open disgust! I was utterly nonplussed, as were many of the other attendees (who, I might add, were very friendly and sincerely interested in the presentation). This kind of thing had never happened to me before. Why would anyone get so upset by my offering to give them my personal attention at the next available opportunity? I could only wonder. Was he some competitor's plant, perhaps? Was his sole purpose to prove that I was some harebrained Microsoft twit who didn't know left from right, a judgment that could then be passed onto the whole lot of us?

Whatever his intentions were, I recovered from the shock, regained my composure, and finished my presentation. What seemed to be an unfortunate misunderstanding left my mind entirely…until it showed up, that is, in one of the weeklies! After spending a dozen or so paragraphs condemning Microsoft for its utter ignorance of the REAL ISSUES at Object World and for its blasphemous misrepresentation of CUSTOMER'S REAL NEEDS, yada yada yada, the reporter, who had appar-

ently been present at my talk, related the whole incident in gory detail. In reverential tones she praised my disgruntled inquisitor as such-and-such "veteran software architect" from such-and-such widely respected company, whereas I was described as—and I quote verbatim—the "arrogant young Microsoft nerd."

Double ouch. I felt like I'd been stabbed. As I read this public mockery, which I knew had also been read by several hundred thousand other people, my heart sank to the middle of my stomach and tied itself in a tight knot. I was young, that was true, still a mere 25. I did work for Microsoft. And yes, perhaps I was something of a nerd (*who* at that show, pray tell, was not?). But arrogant? There wasn't a trace of such a sentiment in my heart! I was simply doing my best to joyfully share what I knew and what I loved.

Instinctively, perhaps, I wanted to fight back in some way, to clear my name and perhaps even humiliate this—as I referred to her at the time, so pardon my saying so—"bitch reporter." But what could I do? Any rebuttal or retaliation would take weeks to see print (if at all) by which time the whole matter would be dusty history. And any personal confrontation would be utterly useless. So I had no other choice but to just sit in my office and eat my humble pie.

Sensitive to my suffering, several of my more sympathetic co-workers stopped by to console me. The bolder, self-assertive types came too, congratulating me for being perhaps the first Microsoft person whose existence was openly acknowledged by the Object Technology crowd. For years it had seemed strictly verboten to mention Object Technology and Microsoft in the same paragraph—so I had to admit that this was something of a success! Nevertheless, it still hurt. It wasn't just that I had been insulted—that would have been easy to take. It was that everything that formed my very self-identity—my career, my

work with OLE, and my very desire to help others—had been dishonored and disgraced. It was in this, in the persecution of what I treasured most, that I suffered.

Interestingly enough, it was about this same time that I began, without really being conscious of it, to explore the lives of others who had undergone persecution; a part of me needed to understand what this sort of suffering was all about. My extracurricular reading (see Chapter Five, pages 69-71), which was in full swing by now, took me into histories of World Wars I and II and the Nazi Holocaust. I read about early Christian "heresies" such as the Gnostics who met their destruction at the hands of the orthodoxy. I also read a somewhat bitter book about the injustices toward women—like the Inquisition and witch-hunts—that have left many an ugly scar on our civilization. In all of these I found inspiration to face my own difficulties with courage; I was especially moved by those who had remained loyal to their deepest inner convictions even when it cost them their freedom or their lives.

I also began to understand how in the very heart of worldly trials was an enormous potential for growth. In so many cases I found that persecution had made people inwardly *stronger*, not weaker. It was as if in being stripped of their narrow external sense of self they discovered a much more expansive identity within; through intense challenges to their assumptions about life they developed a transcendent vision of reality and an understanding of what was *truly* meaningful.

By a fascinating coincidence—or perhaps by design!—I was asked to do something of this nature for OLE shortly after my Object World experience. While OLE had gained considerable support in the industry it hadn't yet found its proper place within Microsoft's overall strategy: its "positioning," as we called it. Being the person most intimate with the technology, this task was assigned to me.

"Where does OLE belong? What is it really trying to accomplish?" In seeking to answer these questions I examined everything that led to its creation. I probed deep into the reasons behind every one of its features and pondered its potential for the future. I followed every possible line of thought, no matter how absurd, to see where it would lead.

And what I found thrilled me—even more deeply than the inspiration behind my book.

As I'll explain more fully in Chapter Fourteen, I discovered (among other things) that OLE had the potential to fundamentally change how software is both created and used. Through OLE, computers could become far more usable than we thought possible. In particular, I postulated an environment where the computer only needed to ask: "What would you like to do?" Once the user told it, the computer would assemble the appropriate application to achieve the desired goal. And if it missed the mark, it would be a quick and simple matter for the user to refine his or her response to get the desired result.

I also discovered that whenever such strides had been made in the past, that is, whenever people had been empowered to create their own solutions rather than waiting for someone else to do it for them, the overall computer industry experienced tremendous expansion. If positioned carefully, OLE could thus become a catalyst for such an explosion, leading to so much growth in the marketplace that no one company could expand fast enough to fill the holes. This meant that while existing companies, both big and small, would prosper, there would also be countless opportunities for entirely *new* companies to sprout and flourish. Literally everyone could be a winner!

I was deeply moved; so deeply, in fact, that within a few months I had completely revised and expanded my book to reflect these insights (later released in May 1995 as *Inside OLE 2nd Edition*). I also wrote my new inspirations in a paper called

"What OLE is *Really* About" which has been described as "perhaps the best paraphrasing of the true purpose of [OLE] that has been published…." In my mind, this paper was certainly the best thing I'd ever written on *any* subject.* And the presentation that I later gave under the same title to an audience of over 1,500 was my best and most inspiring talk ever.

Thus through the persecution I endured on OLE's behalf emerged a vision that transcended my own needs and goals. For that matter it also transcended the needs of my workgroup and even those of Microsoft.

Yes, I could now see OLE's true significance.

I had found its Grand Purpose.

I had understood its place in the Great Scheme of Things.

And, by extension, I gained more insights about my own.

As I said earlier, my identity was intertwined with OLE. In exploring its life purpose I naturally began to seek answers to the fundamental questions of my own existence. (Indeed, I almost once wrote a paper called "OLE and the Meaning of Life.") The voluminous entries in my 1994 journal give testimony to the extent of my search. What is life for? What is it all about? Where do I fit in? Again and again these questions returned to

---

* It still circulates in the developer community to this day and can be found on msdn.microsoft.com by searching on the title. The quote here comes from an "Under the Hood" column by Kevin Gordon that unfortunately no longer appears on the Internet. I'm also highly amused by a 1997 article called "Oils of OLE" by Eric Binary Anderson that ended with this tribute: "[What OLE is *Really* About] is the clearest and most complete summary of OLE technologies you'll find. Brockschmidt seems to believe that Microsoft developed OLE as a gift to the development community to engender competition and to keep Microsoft from driving the industry. Kraig must have sold some snake oil in his day, though, because I came away convinced."

my mind, creating a magnetism that brought me increasingly in touch with others who were also seeking answers.

In particular, I became actively involved with a Microsoft email group where this sort of metaphysical stuff was actively bandied about (see Chapter Thirteen). Through this group I met Richard Brodie, an ex-Microsoft programmer who wrote the original version of Microsoft Word many years ago. Despite more than a decade of professional and financial success with Microsoft, Richard realized one day that his life was spiritually empty. Seeing this for the problem that it truly was, he left Microsoft and spent several years doing everything he could to climb out of his hole, attending dozens of personal growth seminars and becoming a dispassionate observer of his own hard experience. In the end, he managed to get some bearing on what was meaningful and important in his life and shared his experience in his book, *Getting Past OK*.

In this book (which Bill Gates described as "incredibly useful!") he offers a series of exercises though which you find not what you *think* will give meaning to your life, but what already *is* most meaningful based on how you have actually lived thus far. The idea is that what we actually *do*, not what we say or think, alone demonstrates what is most important to us and what we truly believe within our innermost hearts. This, you can even say, is our only true and personal religion.

At the end of Richard's process you have what he calls your "Success Checklist." These are the things in your life—your "core needs"—that really matter to you (such as sharing joy) as distinct from the *means* of fulfilling those needs (such as writing a book) and the *structures* that support those means (such as having a writing job or a good source of ideas). To fulfill your core needs, Richard says, through whatever means and struc-

tures are appropriate, is your life's purpose.*

I first read *Getting Past OK* in the middle of 1994 and worked through its exercises toward the end of the year. My list of core needs came out as follows:

- Wholeness, integration, and connection with all life; a state of total peace with the universe;
- Effectiveness, or doing what is right and what will lead to the greatest growth and happiness;
- Freedom and unboundedness, not being restricted by limitations in thought or belief, having the willingness to try things no matter how crazy they seem;
- Awareness and complete knowledge—that is, *wisdom*—and
- Love, expressed for all and deeply felt within my own self.

Yes, I had to admit, these were the things that really *were* important to me—they were what I had always been seeking. They had been guiding my life from behind the scenes through every storm and success alike. And now that I had them spelled out so clearly, I tried to consciously keep them in the forefront of my mind, seeking to deepen my experience of them wherever I went. As I did so, my awareness was gradually transformed, setting the stage for what was the next critical step in my spiritual growth.

---

* The only thing I feel is missing from Richard's book is that once you have your Success Checklist he doesn't advise you on how to work backwards to find appropriate means and structures. I'm working to fill this hole in a book called *Finding Focus*. In any case I was, and remain, grateful to Richard for his work.

No wonder the great master from Galilee, quoted at the beginning of this chapter, encouraged us to pray for and bless those that would persecute us. Without tests and trials we might never discover that which gives our lives purpose, nor the strength within ourselves to actually live that purpose. These are blessings worthy of our deepest gratitude.

CHAPTER THIRTEEN: AN INTERMISSION OF SORTS

A Flick of the Switch

Lord, make me an instrument of Thy peace.
Where there is hatred, let me sow love.
Where there is injury, let me sow pardon.
Where there is discord, let me sow unity.
Where there is doubt, let me sow faith.
Where there is error, let me sow truth.
Where there is despair, let me sow hope.
Where there is sadness, let me sow joy.
Where there is darkness, let me sow light.

—Saint Francis of Assisi

"You have 672 new messages." Long before "spam" even existed it was not uncommon to see this sort of alert when we checked our morning email. Since its early years, Microsoft's lifeblood—its primary means of internal communication—has been email. Millions of messages course through its veins every day. From the senior executives down to the folks on the manufacturing floor, in the mailroom, and in the cafeterias, *everyone* at Microsoft has email.

It was entirely possible to spend your whole day doing nothing but reading, responding to, and deleting email messages. No matter who you were you could expect to receive at least thirty meaningful messages every day—if not several hundred!

Many people received even more. I remember a friend of mine once having nearly two thousand unread messages in his inbox—and those were only the ones marked "urgent"!

Despite this torrential flood of messages, we all loved email: it was integral to our work rather than an annoying distraction. It was vastly more efficient than paper mail and wasn't disruptive like the telephone. It was also usually much more efficient to send email than to find someone by phone or in person—oftentimes a co-worker would call or come by your office only to say "check your email." What's more, people who happened to be out of town usually checked their email several times a day while outright ignoring their telephone voicemail.

One of the best features of our email system, and the major reason why we got so many messages, were the "group aliases." An alias is a single email address that automatically maps to any number of other addresses, including other aliases. The *drg* alias, for example, mapped to everyone in Developer Relations; *sysmktg* to all of Systems Marketing. We had aliases, in fact, for every part of Microsoft's organizational structure.

With this powerful feature you could send a single message to a single alias and get it out to hundreds or thousands of people without needing to know individual addresses. This was especially true for the *msft* alias, the one that sent mail to *everyone* in the company!

Group aliases also allowed us to form various discussion or information groups irrespective of organizational boundaries. The *olecore* group, for instance, included everyone who was deeply involved with the OLE technology no matter where they were in the company or, for that matter, the world. There were also aliases for varying groups of vice presidents, program managers, administrative assistants, and so forth. If you could think of any reason why you might want to email a particular group of people, there was usually an alias for it.

Our discussions were not restricted to company business. There were group aliases for everything from home buying and car maintenance to bungee jumping, punk rock, Dungeons & Dragons, and every religion known to modern man. Name an interest and there was an alias for it. And all it took to join a group was—what else!—sending a piece of email to the alias administrator. You could then look forward to even more messages in your inbox every morning!

A fun part about the whole thing was that all aliases were equal: the email system never asked if you *really* intended to send a message to a large group. So messages that were meant for only one or two people occasionally got sent to many, many more. One day, for instance, a woman sent a very loving (G-rated) message to her fiancé (many couples within Microsoft take care of family business in this manner). Whoops! She accidentally sent the message to an alias with four hundred members! Her message was so sweet, however, that it charmed everyone who read it. Soon her own inbox was full of congratulations on her upcoming wedding and many other words of support. And the whole incident was so touching that it eventually made the back cover of *Micronews*, our weekly company newsletter.

The group alias involved was one to which I belonged. Its name was *soleil* (French for "sun") which stood in this context for "Sharing Our Life Experiences Is Loving." Its unofficial name was the "personal growth alias" and was where people discussed things like psychology, spirituality, metaphysics, inspirational books, alternative medicine, yoga, tai chi, meditation, ecology, charitable works, UFO's, and whatever else you care to imagine. Indeed, *soleil* represented such an unorthodox menagerie of subjects that I referred to it as the "weirdo alias." But that's what made it fun! We're all weird in some way or another—why not enjoy it?

Conversations on *soleil* were always interesting, uplifting, and rich with attitudes of compassion and open-mindedness. In this loving and supportive environment, individuals commonly asked the whole group for recommendations of some kind—a good naturopath, a nice place to stay at the ocean, an honest mechanic, or a non-profit organization that needed volunteers. And because everyone in the group habitually approached each new message with an open heart, there were often a dozen or more responses to such requests within an hour or two.

On February 1st, 1996,* about 11:30am, one of our group's most active members sent this message:

> Can anyone recommend a divorce lawyer who knows about dealing with Microsoft stock options? The couple involved don't want to have to cash in the options for the non-Microsoft partner.

Not surprisingly, a reply came within minutes—addressed to the whole group. But it wasn't a helpful recommendation, it was a *scathing* condemnation! Though very short in and of itself, the message was essentially a righteous tirade on the evils of divorce and its sole responsibility for dysfunctional families and everything else that's wrong in the world. It also held a strong tone of judgment against the couple themselves for even *thinking* about separation!

"Whoa! Where did *that* come from?" I thought, startled. *Soleil* was founded for sharing *love*—it was a complete shock to see such negativity!

Again, the message was but a few short sentences. Nevertheless, it had this ENORMOUS negative power! You could *feel* its anger: after reading only a single sentence I had this sudden, sickening, sinking feeling in my gut: *something was very*

---

* Chronologically this story is out of sequence in the book. I place it here as a brief respite before the next chapter.

wrong. "Run away! Run away!" my mind screamed. But it was already too late. I had opened the message with my usual receptivity and before I knew it, I found myself infected with terrible emotions.

And it wasn't just me…everyone else who had read even a little bit of the message had become infected themselves.

So of course you know what happened next: "the battle was joined" and the bombs began to explode. Within only a few minutes a barrage of counter-attacks assaulted our inboxes as every reply was sent to the entire group. And then came the counter-counter-attacks. Then the counter-counter-counter attacks! Minute by awful minute people were taking sides and jumping into the brawl. My inbox was literally inundated with new responses.

Responses? Hardly…they were tirades. They were venomous maledictions. They were hellfire and damnation! And, of course, everyone believed that Truth was on their side: there was no hope for even so much as a cease-fire!

It was incredible to witness how quickly the whole thing got out of control. Cherished opinions were being attacked and the natural reaction was to fight back. Victory, as the combatants seemed to believe, was a matter of who had the biggest gun, the strongest fist, or the loudest voice. Anger beget anger; insult beget insult! And the bright land of *soleil*, once flowing with the sweetness of milk and honey, was now plagued with darkness and bitter poisons. Love was nowhere to be found. Even the few crying pleas to stop the bloodshed were corrupt with anger and negativity: they succeeded in only escalating the carnage further.

A great spiritual teacher once wittily said that "you can't beat the darkness out of a room with a stick…." Well, everyone seemed hell-bent on proving this principle wrong. The man who started the whole thing was trying to beat out the darkness of

divorce with a stick of condemnation and righteousness. In response, others were trying to beat out the darkness of his condemnation with their own sticks of judgment. And as the war intensified, others picked up sticks of reason, sticks of emotion, and even sticks of compassionate understanding. But no matter what the motive they were still sticks, they were still used for beating, and none of them were doing any good whatsoever.

I was right in the thick of it all myself, fully ensnared by anger and fury. Within five minutes I picked up my own stick of self-righteousness and started writing—or, more accurately, SHOUTING—my own declaration of war![*]

Then I caught myself. Mustering all the willpower within me, I stopped cold. "It won't help one bit," I told myself, "to throw any more fuel on this fire." I just said NO. Dropping my stick, so to speak, I cancelled my message, purged my inbox of everything else, and took a nice, deep breath.

Aaah. I immediately felt as though the mud had been hosed off. I felt cleansed and relieved. Now I could just ignore the raging battle and get back to my work.

Whoops! Not so fast, my friend. When one is caught in the middle of mud-slinging you keep getting dirty no matter how many times you wash up. The email kept coming. Each message brought a fresh burst of negativity that scorched me before I could even delete it.

No matter how much I tried to pretend otherwise, darkness was penetrating my entire being. I had turned away from anger in hope that it would just go away and leave me alone. But it wouldn't. As the saying goes, the road to hell is paved

---

[*] YOU SHOUT IN EMAIL BY WRITING IN ALL CAPS AND USING LOTS OF EXCLAMATION MARKS!!!!!!!

with good intentions. With an almost tangibly conscious force, the darkness kept pulling me downward: physically, that knot in my stomach became even tighter; mentally, I couldn't concentrate; spiritually, I felt crushed.

Nor was I alone—I realized that with only two or three dozen members of *soleil* taking an active part in the carnage, hundreds of others were enduring a silent agony. And they probably felt like I did, just hoping to stay out of it long enough for the anguish to subside.

Then I had a horrifying thought: the negativity was so powerful that its destructive vibrations would spread into everything I did for the rest of the day, perhaps even for another week. It would spread into my work, into my relationships, into my very thoughts! And if hundreds of others…a chill ran down my spine. We had each become an unwilling carrier of a dread disease. Our repressed anger and bitterness would ultimately infect our families, our friends, and everyone else we came in contact with. They would, in turn, infect others, who would themselves infect more. And—<shudder>—a further ghastly realization surfaced: it wasn't just happening at Microsoft's corporate campus. *Soleil* had members in many of Microsoft's nationwide sales offices and foreign subsidiaries—the epidemic was global!

This just couldn't go on. *Something* had to be done. But what? I asked the question to myself over and over: "What can I do? What can I do? What power do I possibly have that can overcome such darkness?"

As I made this desperate inward search a certain thought took shape in my mind. Despite its many challenges in the marketplace and the courtrooms, I couldn't remember a time when Microsoft had responded with hostility or malice: *Microsoft overcame negativity with an even greater amount of* **positive** *energy*. No matter what the situation, we countered every

attack with an even greater determination to succeed, holding fast to our highest ideals.

The downward path of negativity and criticism is always easy: all you have to do is fall. It takes great energy and courage, on the other hand, to stand up and live your ideals—or to simply be positive—especially when no one else seems even willing to try. It is not a path for weaklings! But simply by making the effort we attune ourselves with Goodness itself, allowing the Divine Light to shine through us and drive the darkness away. In this we each have the power to change the world, if we would but choose it.

Yes, that was the answer: *you can't beat out the darkness with a stick, but you can turn on the light!* I had to turn on the light. I had to express some kind of positive energy that was more powerful than the downward pull of the ongoing war.

Deeply inspired by this thought, I recalled why our group had formed in the first place—*Sharing Our Life Experiences Is Loving. Soleil* was a vehicle for light of every shade and hue, which together made the loveliest rainbow.

Over two years with the group I'd saved various touching stories, instructive jokes, and profound quotations. While my inbox continued to swell with putridity, I read through all of these gems and picked out a few of the best. One was this passage by Marie Dominique-Ellis, *soleil's* founder:

> *Be compassionate*
> *Allow people to be who they are*
> *Allow people to express what they think*
> *Allow yourself to not take things personally*
>
> *If someone does not play the game*
> *According to the rules*
> *Let's give them the rules*
> *Instead of raising our fists*

Another told the fun episode from Sesame Street in which Oscar the Grouch was trying to spread grumpiness at Christmastime by giving away what he thought were useless and insulting gifts. But in each case the recipient found the item most helpful and Oscar only succeeded in spreading joy! Another story told of two Arabs who were driving cars in the open desert and collided. Instead of getting into a fight, however, they embraced each other. "Allah be praised," they cried, "for if we hadn't crashed we would never have met each other!" I also found the original account of the woman who had accidentally broadcast that charming message to her fiancé.

As the intense battle of negativity continued in unabated fury, I mustered every ounce of love and courage in me and composed a message with these stories. Here is how it began:

> From: kraigb
> To: soleil
> Sent: Thursday, February 01, 1996 12:23 PM
> Subject: Bringing us back to center...
>
> Time and time again, words that appear on SOLEIL have had the power to drastically affect those who read them. Recently an outpouring of love regarding an accidentally broadcast message made the back cover of Micronews. Negative words can also have tremendous effect, and on this alias can sour a day for hundreds of people. This mail is my own personal attempt to turn anger into love.
>
> In light of the current exchange on this alias, I'd like to share a few pieces I've picked up and saved from the last three years, hopefully in order to bring us back to that stable center where we can love, respond, and learn from each other, in the spirit of SOLEIL: Sharing Our Life Experiences Is Loving.

With my heart racing nervously, I wondered how people would respond. Would anyone notice? Would they turn their anger on me? I just didn't know—whatever the risks, I simply had to try.

I sent my message…

Instantly I once again felt cleansed, this time permanently. This strong, positive expression of love and joy had reversed the flow. No longer were black tentacles of hatred reaching out of my inbox to strangle me—the Light drove them back for good. I knew that the war could no longer touch me. My queasiness left me completely, my mind was suddenly clear, and my soul was all at once uplifted. Wordless prayers of gratitude rose from my heart. Never before had I experienced such an instantaneous healing.

I came that day to appreciate both the incredible power of negativity and also the even more incredible power of love. I also came to a clear understanding that while it's not up to us to *create* these powers, we choose which one flows *through* us. Will we be instruments of darkness, or instruments of light? This is really the only choice we have. It is the only real power we have.

What we choose to express, we become. To be loving simply means to choose love rather than anger or hatred. To be joyful simply means to choose joy rather than sorrow.

May we thus each pray with Saint Francis: "Make me an instrument of Thy peace."

Oh yes, the response to my message? It was truly miraculous: the whole energy of the situation completely inverted. Whereas my inbox had been filling up with messages of anger and hatred, it was now filling up with only messages of love, joy, and gratitude—broadcast, as always, to the entire alias. Dozens of people said how my one little message had cleansed them as I had been cleansed. One woman wrote: "*Thank you. You saved me from sending a very angry flame to this person. Flames were issuing from my fingers as I typed!!*" Others appreciated the reminder of *soleil's* purpose. Some simply enjoyed the uplifting stories. And one man who had just joined the

group the day before told us all how in the midst of the battle he was really wondering what he'd gotten himself into! But now, having witnessed this undeniable transformation, he understood both the group and the Power—with so many different names—that gave it life.

And the miracle continued. When I arrived at my desk the next morning I was overwhelmed to tears. For there, in my inbox, were replies not just from Microsoft employees in the Unites States, but from all over the world, every one of them bursting with sweetness and joy. What could very well have been a virulent scourge of worldwide anger had been transmuted into a global epidemic of Light.

In fact, from the moment my note appeared in everyone's inbox there was only one more negative message. It was from the same man who had issued that first scathing reply.

"This alias sucks!" he screamed, "I'm leaving it for good!"

You can't beat out the darkness with a stick,
but turn on the light
and the darkness will vanish
as though it had never been.

CHAPTER FOURTEEN

Breakthrough

One of my favorite books—out of the two hundred or so that I read during my years at Microsoft—was *The Chalice and the Blade* by Riane Eisler. It presents convincing archeological evidence that the ancient civilization of Crete was not of the barbaric and primitive variety that we normally assume for its era but rather one that was more culturally advanced than our own. Though lacking in technology, Crete enjoyed harmony, peace, and joy as the norm. In comparison to our hectic world and all its so-called "conveniences," Crete's legacy challenges us to reexamine our fundamental principles about what life is and how best to live it.

Within this context Eisler presents her Cultural Evolution Theory. Every so often, she says, there comes a time when a civilization must make a critical choice: shall it be based on principles of domination or shall it be based on principles cooperation and partnership? With plenty of historical evidence to back her claims, Eisler demonstrates that choosing the dominator model invariably leads to collapse whereas choosing the partnership model leads to new cultural advancements. She then goes on to show rather persuasively that we are facing just such a critical juncture in our own era: we ourselves must make the choice.

When I read the book in early 1995 I was deeply inspired—its profound ideas offered an expansive scope for my efforts to describe OLE's "Life Purpose" (see Chapter Twelve). In particular, I clearly saw how our broader culture's critical juncture was reflected in the software industry. Open-source projects aside, today's software industry is mostly based on domination: millions of users are essentially at the mercy of a few powerful software companies, Microsoft being the foremost. Although these companies take enormous pains to serve customer needs, most consumers have little or no direct influence on the software that they're more or less compelled to use. That is, they really don't get software that works how they *personally* want it to work: they have to do things the way the *software* wants it done, no matter how many focus groups and customer studies went into its design. This is the fundamental reason why so many people find computers frustrating and annoying.

This problem is a natural outgrowth of the way we learned to build software in the first place. Way back in the annals of computer science we find that a computer application was a program designed to solve a very specific problem. The earliest computers, in fact, were hard-wired to do one thing and one thing only, like deciphering encryption codes during World War II. "Programming" back then was, in fact, an integral part of a computer's physical construction.

Then someone came up with the basic idea of an operating system—a layer that isolated programs from the specifics of the computer's hardware thereby allowing you to load and run any number of distinct programs on the same physical machine. Programming now became a completely separate field and created, as a result, thousands of opportunities for specialized "software engineers." When someone wanted to use a computer for a particular problem they gave a detailed problem description (the specifications) to some programmers who then

created a specific solution to that problem: the application. When the application had served its purpose it was archived or tossed out altogether; only a few of these programs could effectively be applied to other problems.

The only difficulty with this approach was that it took a long time to get problems solved—it could take years before an application was perfected. So people began looking for ways to solve multiple problems with one application. Gradually they developed general-purpose programs that each solved a *class* of problems rather than a specific one.

This was another tremendous boon for the computer industry as a whole. Suddenly you could go to a store and buy an off-the-shelf software package and adapt it to your particular needs. I stress that word "adapt"—a word processor only knows about the abstraction of a "document"; it really doesn't know about things like a resume or a letter to grandma. A spreadsheet program only knows about pages of interlinked cells—it doesn't really have a clue about things like ledgers and balance sheets.

This means that the users of that software are left with the task of adaptation: they have to bridge the gap between their specific needs and the software's particular abstraction. End-users have to create the document templates, the relationships between cells in a spreadsheet, and all the tables and queries in a database.

This process of mental mapping is exactly what computer users find the most difficult. In response, software companies like Microsoft have done their best over the years to make things easier, even magnificently so. The most notable developments are the templates and "wizards" that automate thousands of specialized tasks—run a wizard, answer a few simple questions, and the computer does just about everything else. In addition, hundreds of customization features—including built-

in programming languages like Visual Basic—have brought us closer than ever to having true applications (problem-specific solutions). And all this has brought even greater prosperity to the industry as a whole.

But it has come at the cost of skyrocketing complexity: our software keeps getting bigger, slower, and more expensive and time-consuming to produce; sophisticated programs are increasingly difficult to maintain with a high degree of quality; and despite all the help from wizards and whatnot, end users actually have to understand more abstractions than ever. (At least now they can hire consultants—yet another specialized profession—for this purpose.)

As I reflected on this state of affairs I just couldn't see how it could go on too much longer. We've been fortunate that computer hardware keeps improving at the rate it has (faster processors, cheaper memory, larger hard drives, etc.).* But can it go on forever, or even another half-century? I wasn't so sure: someday the sheer complexity of it all would have to collapse under its own weight…unless… unless there was some kind of fundamental shift at the very source of the problem.

This is where I saw OLE coming into its own. Rather than just adding more fuel to the fires of complexity, OLE brought a new simplicity. It made it possible to build software in a new way, the catalyst for a real breakthrough.

Traditionally speaking, applications were generally built as a single, massive monolith with everything packaged into a single executable (EXE) file. Over time it became desirable to break out parts of those programs into stand-alone modules

* Microsoft programmers were sometimes challenged on the sluggishness of their programs when run on the most common computers of their day. "Don't worry!" was a usual response, "In a year the hardware will get so much better that you won't notice any longer."

that could be shared with other programs, thereby reducing the size and complexity of them all. Technologies like OLE very much facilitated the process. Nevertheless, many applications are still built around a fundamentally monolithic architecture.

OLE introduced a different fundamental architecture altogether: the ability to create complex software from discrete components. Rather than sharable modules being used merely as add-ons to larger monolithic programs, those programs could themselves be built with such components from the ground up. As a result, the enormous processing power available to us today could be focused much more precisely on those features that are actually being *used* rather than committed to an ever-increasing array of highly-specialized features that many users never invoke.

An associate of mine at Microsoft, Crispin Goswell, demonstrated this potential by assembling, from small OLE-based components, a program that offered most of the same features as Microsoft Word—one of the largest and most complex personal computer programs in the world—yet was significantly smaller and *much* faster. It was so fast, in fact, that when Crispin scrolled through a thousand-page document with blinding speed I didn't believe his program was actually reading information from the hard drive.

Even more impressive was the fact that each of his components were, in themselves, relatively simple: one person like Crispin was able to understand, develop, test, and maintain everything himself. In comparison, a complex application like Word demands a large team to build and maintain it.*

---

* Crispin, of course, had the luxury of starting from scratch, which was not and is not a practical option for the Word team. Nevertheless, Word—and Microsoft Office in general—is gradually becoming more and more "componentized" with each version, allowing some very powerful features (like

The possibility of such dramatic progress—in both speed and simplicity—was utterly thrilling to me: OLE had the potential to truly revolutionize how software was built. But it went even deeper than that—the more I thought about it, the more I realized that it could also revolutionize *who* could build applications in the first place.

Consider this thought: systems built on a reasonably large number of simple components allow for significantly more combinations—and are thus far more flexible and adaptable—than those based on a small number of complex components. The ninety or so stable chemical elements of the periodic table can be combined into billions of wondrous forms from the simple to the elaborate. If, on the other hand, our "elements" were a Mickey Mouse telephone, a cheetah, a stalk of broccoli, and a Saturn V moon rocket, the world would be a pretty dull place.

The technological future I saw with OLE's component model, then, was one in which many smaller software companies—even individuals—would produce tens of thousands of highly diversified building blocks as opposed to having only a few large companies like Microsoft produce a mere handful of general-purpose megaliths. With these components it would be a simple matter, as Crispin had already proved, to construct many of today's applications without any loss of functionality. At the same time, those very components could be used to build literally *millions* of other special-purpose applications more quickly and more economically than ever before.

This was the key: people's real needs would be served better than ever, making computers far easier to use and thereby far more attractive to employ in one's day-to-day activities. This

---

speech recognition) to be added without any significant performance penalty to the application's core functionality.

would, in turn, catalyze a tremendous expansion of the entire industry, and within that expansion it would be logistically impossible for any one company, even Microsoft, to grow quickly enough to fill in all the gaps. While big software companies would continue to thrive there would also be innumerable opportunities for many smaller ones. Everyone, in short, could emerge a winner!

And if this wasn't enough of a boon itself, I figured that with enough components, profitably produced, it would become possible to truly build exact solutions to specific problems—true applications—*virtually on demand*. Users would no longer have to wait months or years for new programs or feature updates: their needs could be fulfilled within weeks, maybe days. I could even see a time when users wouldn't have to wait at all. With likely advances in speech recognition and the ability for a computer to actually *understand* what you say, there would come a time when you could simply tell the computer what you wanted to do and the exact application that met your specific needs would be built on the spot. All that tedious mental mapping would be completely eliminated.

Power would thus shift from the big software companies to the individual computer user. Diversity, not domination, would be the rule; cooperation—critical to the interoperability of components—would replace cutthroat competition.

My heart overflowed with joy the more I thought about such possibilities. The sheer simplicity of this new paradigm would be not only wonderful for technology but also wonderful for all the *people* involved. And not just for end-users, mind you, but also for programmers—building small, focused components was far more enjoyable than working on some obscure part of a huge monolith. What's more, cooperation just *felt* better. If there's one thing that's always saddened me about the software industry it's the degree of selfishness and greed that I've

encountered in both individuals and corporations. Anything that might encourage people toward generosity and self-expansion was, to my mind, a worthwhile cause.

"Now isn't that a fascinating thought!" I reflected. "What would be the effect of this fundamental shift on the hundreds of millions of people who might use computers in this new way? If people knew, even vaguely, that a cooperative approach had produced the best solutions to their technical problems, would they perhaps begin to approach the rest of their lives in a more cooperative manner as well? Would they learn, in day-to-day living, to choose cooperation over selfish competition? Would they decide, in fact, to choose partnership over domination?"

If so, this simple technological shift could very well be a deciding factor in Eisler's cultural breakthrough. *OLE's "Life Purpose" was thus a very deep purpose indeed!*

Now you should know that this whole thought process was a somewhat drastic extension of the job I had been given at Microsoft to find OLE's "positioning" (see Chapter Twelve, page 166). Even when I shared only the strictly technological and economic aspects with my closest associates, I definitely found myself pushing their limits of comfort and convention.

Nevertheless, I could not ignore such extraordinary possibilities. I was increasingly searching for meaning in my life as a whole and I needed to find meaning in my work. I simply wasn't doing it to just have fun or to make money: I needed a high aspiration, something *universal*. Without some sense of meaning, life seemed a futile exercise filled with cares and worries over endless trivialities.

I found meaning in my vision of OLE: the more I worked with it the more it seemed a doorway through which I might even find the answers to certain deep philosophical questions.

As I mentioned in Chapter Twelve, I almost wrote a paper called "OLE and the Meaning of Life." In any case, doing my utmost to make this vision a reality, even if it took the rest of my career, was the most worthwhile and inspiring activity I could think of. Indeed, I thought of it as my very *responsibility* to the world and to Life itself.

And Life seemed to respond favorably with new opportunities that supported my resolution. First, I needed to update my book to cover a few things I had omitted before and to include chapters on some more recent additions to the OLE technologies. In the process of editing the whole book again I tried to align even my most detailed discussions with my overall vision. Each chapter led step by step to the last one in which I articulated my dreams for the future. And so there it was—when this new book, *Inside OLE 2nd Edition,* came out in May of 1995, I had committed my aspirations to print.*

During my work on this second edition came another timely opportunity: one of the original OLE architects asked me to join the OLE design team. His invitation couldn't have come at a better time—Developer Relations was being reorganized, and because OLE no longer needed much evangelism there was talk of having me do the same thing all over again with other upcoming technologies like Interactive Television. Given that I didn't even *watch* TV and that it was OLE that had really won my heart, it was easy to leave DRG. In March of 1995 I joined the OLE Program Management group where I hoped to guide the technology toward my highest goals.†

---

* By then, we had dropped the version number from "OLE 2" and officially referred to it as just "OLE."

† OLE Program Management was, ironically, part of the Windows NT project, the one that I would have ended up on had I gotten that program management job with OS/2 way back in 1990 (see Chapter Five).

A third opportunity came through my public status as the "OLE guru" (which had, by now, become so focused on me, personally, that people pretty much forgot that I stilled worked for Microsoft). Invitations to give lectures continued to pour in, each one giving me a new platform from which to share my ambitious vision. One such engagement, the Visual C++ Developer's Conference (Santa Clara, California, June 1995), came right after the publication of *Inside OLE 2nd Edition*. It was a golden opportunity in this regard—to a receptive audience of over fifteen hundred I gave my talk called "What OLE is *Really* About." Again, this talk was easily the best technical talk I've ever given, sparkling with joy and wonder throughout all ninety minutes.

The presentation was a complete success—people were both inspired by what I had said and seemed to intuitively understand what I was driving at. This, combined with everything else, gave me good reason to feel that I was onto something of real importance. With coordination, cooperation, and certainly a lot of hard work, I felt wholly certain that we—as an industry, not just Microsoft—could transform the way software was written, improve the lives of millions, and uplift the very consciousness of the planet. More than ever I resolved to dedicate my energies to this purpose.

I had driven down to this conference in Santa Clara from my home near Seattle so I could enjoy a long, peaceful drive up the Coastal Highway as a little vacation afterwards. First I visited my dear friend and fellow author/lecturer, Bruce Eckel, in a small town in Marin County. Then I enjoyed the expansive beaches of Point Reyes before winding my way northward to Humboldt State Park and some silent communion with the

ancient redwoods. Farther north, at Arcata, I spent an inspiring sunrise with the residents of a bird sanctuary. On into Oregon I took the opportunity to visit a wild game park where one can mingle with, touch, and even hug an assortment of animals including goats, llamas, and deer. And before heading home I spent several quiet days in oceanside solitude at the small town of Yachats (pronounced YAH-hots), where incoming waves get funneled through narrow, rocky channels and burst into the heavens like the geysers of Yellowstone.

In removing myself from my normal routine, and from the influence of people in general, so many little concerns melted away. Speech became unnecessary; silence reigned. The restless thoughts of my mind slowed considerably, even coming to a complete halt now and then, allowing me time to calmly reflect upon my present life and the future to come.

During this seclusion, a deeper part of myself got an all-too-rare chance to speak. Above the soothing background hum of the ocean, my soul reminded me that something was missing. Was it certain material possessions or extravagant vacations? No, my wife Kristi and I had everything we wanted and the wherewithal to acquire anything else we might desire for years to come. Was it that we had recently decided not to raise a family of our own? No, neither one of us felt drawn in that direction at the time. Was it an absence of values that were important to me? No, these I had uncovered with the help of Richard Brodie's book *Getting Past OK* (see Chapter Twelve). Love, wholeness, an expanded awareness, connection with life, openness to truth…these essentially defined my personal religion. And I could clearly see how my future work with OLE would provide many means to fulfill those needs.

So what was missing? Finally it became clear: deep down, at the very center of my soul, I realized I was lonely. There was an emptiness that no outward relationships, possessions, or

ambitions seemed able to fill. It was a very real sense of how transitory everything in life truly is. Whether stolen, lost, or destroyed in fire, everything I owned would someday be taken away—by death if nothing else. So too with my relationships—someday either my wife or I would pass away before the other. Children (as have entered our lives more recently) would grow up and leave home without any guarantee that they'd "keep us alive" in them. Parents, friends, teachers, celebrities—countless others!—they would all vanish from the stage of time. So would any company, any job, and any career—including every technological dream I had for OLE. All that would remain is what's *inside,* the consciousness and love dwelling at the center of my own being.

And I wasn't sure that there was anything in there at all!

When I returned home in mid-June from my (ha-ha!) "vacation," I was in inner turmoil: on the one hand, I cared deeply about my goals for component-based software; on the other hand, I knew that they would never fulfill the deeper purpose of my existence nor would they be able to fill that emptiness in my heart. I kept having to ask myself, "What truly gives life meaning?" Yes, I had some answers in the form of my most important values, but I had no idea how to really *live* them or how to bring them into the very core of my being. What, indeed, was my "being"? Who was this "me"?

The more I asked such questions, the more I found myself returning out of necessity to matters of spirituality. Did some greater reality—God—really exist? The only time I had really felt sure about this was when I was nine years old and anxiously awaiting a family trip to Hawaii. The night before our departure I couldn't locate my swim mask and prayed to find it. "God must really listen!" I thought, because I found it first

thing in the morning. But over the years I had become increasingly dissatisfied with the whole package, especially as presented by the usual religious establishments. Of what use is a God who needed us to flatter him with praise? Why would God prefer that we talk *about* him rather than *to* him? And how on earth could God be willing to damn us to Hell for all eternity for the slightest mishap if we ourselves are supposed to forgive even the most heinous offenses against us? Such theology really leaves something to be desired.

I had also become increasingly disgusted with so-called "religious" people who presumed to know God's Will but were wholly willing to lie, cheat, and otherwise act in blatantly unspiritual ways simply because it was convenient or profitable for them to do so. What I heard of God's Will was usually, in fact, nothing more than a set of political ideas about who got the money and who should be labeled a criminal.* (Money and power, indeed, are the true deities of our culture.) Nor could I believe that any God worthy of the name, if he existed at all, cared more about how I voted and whether or not I listened to certain radio talk-shows than with the purity of my own heart. I believed in Love and Peace. I believed in Joy and Harmony. Yet according to the judgment of many religious people I knew, I was a prime candidate for hellfire and damnation unless I embraced narrow sectarian views.

One night I felt really ticked off about the whole mess. I openly challenged God and his opposition to clear matters up by sending an angel and a demon, respectively, to hash it all out. "Truth!" my soul cried, "Show me truth!" But all I got was silence…

---

* Laws, after all, don't really eliminate certain behaviors—they merely say whether those who so behave should be punished or incarcerated.

At this point I decided that God rarely, if ever, works in this world through explicit supernatural events. *Good and Evil are states of consciousness*: they operate *from within*, through those who attune themselves to their respective "vibrations." As I'd started to realize a few years earlier, it is our actions—even our very thoughts—that determine, moment to moment, whether we are in the "heaven" of a pure heart or the "hell" of defilement. So no matter what anyone said or told me it simply had to be true that by choosing to express truly good qualities in one's life, like those in which I believed, one became an instrument for God, for Goodness, and that by expressing the opposite qualities, including selfishness, hatred, judgment, and greed, one became a channel for The Other Guy.

I articulated this conclusion in my personal journal on Independence Day, July 4th 1995:

> A human being is the junction between "spirit"—all knowledge and awareness—and the physical material world. Only humans can take understanding from spirit and manifest it through physical action. This is the special providence of humanity, and the ability of each individual human…

We as human beings exist at a unique position, in other words, at the boundary between that which is Spirit—God—and his Creation—Nature. By drawing on God's creativity and inspiration—as I had experienced while writing *Inside OLE 2*—we can manifest him directly in the world and channel his blessings to others. Well, "can" is too weak—"should" is more like it! Bringing God's grace into the world isn't just a possibility, it is our deepest *responsibility!* And it's an entirely *individual* matter: nothing and no one—no church, no religion, no political party, nothing—can define what it means for each unique soul to bring God into manifestation. *One has to experience God directly, within oneself, and express that inner connection according to his or her best understanding.*

That night, as if to confirm these realizations, I had the most extraordinary dream. It banished any doubts about the existence of something I could related to as God. While the specifics of this dream I prefer to cherish in my own heart, what I experienced was, simply said, the combined presence of Love and Beauty in a most divine and complete form. I had never imagined such Beauty. I had never felt such overwhelming Love! Yet this presence was no stranger: it was, in a very real sense, my most intimate and eternal reality.

I was completely given to this presence for the remainder of the dream. Everything else—my relationships, my possessions, my job, my life…each symbolized by some other element of the dream—was utterly unimportant. *Love was Everything. Nothing else, absolutely nothing else, mattered in the least.*

I awoke with that state of consciousness and discovered that this presence of Love remained alive in my heart: the seeds, the ones that God had planted during my experience with *Inside OLE 2* (see Chapter Ten, page 142), had finally sprouted! For it was the same heart feeling that I had first experienced back then, only now more distinct and much more intense. From this point on I could recall that feeling with a mere thought, no matter what I was doing outwardly.

As the weeks went by I found myself joyously recalling this Presence again and again: it became my intimate companion. While it might be shrouded by mists of distractions, I knew for certain that *it was forever within me and could never be taken away*—not by disaster, not by disease, not even by death. It was truly that one missing piece, that one piece to fill my inner emptiness.

Even so, it was a beginning, not an end. There was still a separation between what I knew as "me" and this Love within me. "I" still had to think about it to feel it.

It was a separation I yearned to overcome. I wanted to give

myself to this Love completely, to have it expand beyond the confines of my little heart and engulf my entire being. I wanted to become *one* with it. Yet I had no idea *how*—there were so many energies that sought to pull me away—my thoughts, my subconscious habits...I didn't yet own these. How could I give away that which I didn't own? How could I break down those barriers?

Thus I began in earnest to seek some kind of real guidance for inner development. I returned to all the books I had read and all the ones still on my list. "Somewhere in all of these," I thought, "must exist the Truth I seek, the wisdom that can guide me to my goal." But I hadn't found any one book or author that brought it all together—I had only found bits and pieces scattered across thousands of pages. My only conclusion was that I would have to put it all together myself, and for this I felt I needed more data…*much* more data.

Accordingly, my so-called "leisure" reading became more intense than ever, consuming many fascinating books about all sorts of seemingly relevant topics: dreaming, the brain, memory, cosmology, religion, psychology, archetypes, immortality, cosmic evolution, consciousness, and one that I'd picked up in England called *Human Potential: The Search for that Something More*. While I read, I not only absorbed the material but also scribbled in the margins whatever meaningful thoughts arose in my mind. Often times while interchangeably reading several unrelated books I would discover and record fascinating connections between them. These too I would write down, then continue my reading.

These connections seemed to hold the key. If I read enough books and kept my mind open to their subtle interrelationships I might finally begin to perceive, by a sort of mental triangulation, the central Truth behind them all. I just *knew* that Truth was in there, somewhere. And when I found that Truth I would

have the guidance I sought. I would learn how to free myself from all distractions and by-paths. I would learn how to dissolve myself in that Love.

Now you've probably noticed that I haven't said much about what I was doing at Microsoft this whole time, mostly the latter half of 1995. The fact of the matter is that I was doing very little. I was hardly in my office, "working" from home several days a week. What's more, my new position in OLE Program Management was not what I had expected. Instead of working toward lofty goals I got pulled into an ever-increasing array of mind-numbing details. But since I was still learning the ropes of program management I figured I would just stick it out for a while and see if things improved.

As had happened during my summer internship six years earlier (see Chapter Three, pages 47-49), it was difficult to concentrate on my job. Again and again I found myself drifting away from the mundane concerns of Microsoft and back to my inner search. My heart yearned for answers; my soul yearned for guidance. I just couldn't think about anything else!

During this time I experienced several "book crises," as I called them. These were flare-ups of desperation when I just stayed at home and read, or worked a few hours before leaving the office (I wasn't doing anything particularly important, so nobody missed me). I also got very serious about reading only those books that I thought to have a high "truth density"—I purged my library of extraneous "junk" books, especially cheap fiction and the like that was dense on entertainment but sparse on ideas. Through this intense process of reading and purging I eventually whittled my reading list down to forty remaining titles that together seemed deep enough to provide the keys I sought. Soon, I felt, I would have enough to start piecing to-

gether the thoughts that were buried in the margins of several hundred books and five years of personal journals.

Of course, the question here was still *how*? How would I sift through it all? Certainly I could read and re-read all my entries and find even subtler connections between them. Or perhaps a technological solution was called for—a computer program that could process all these pieces of text and somehow find their common threads.

Yet was it really possible, I wondered, to find Truth like this? No matter how many books I read there would be thousands more I hadn't. Without reading them *all,* could I really expect to arrive at TRUTH and not just a reasonable approximation? And could I really write the kind of program I had in mind? It was so far beyond my present abilities that it would probably take me decades to complete. Still, I just had to find answers! If I had to spend the rest of my life working on such a program, so be it….

Then I began to ask myself if there might not be a more efficient approach to this problem. Faced with such a monumental and uncertain programming task—not to mention the enormous tedium required to read books, mark passages, scribble margin notes, and copy them all into a computer—I wondered whether my own brain was actually far better suited to the task. Would it be possible to develop my own mind to the point where I might *internally* process all the things I've read and all the thoughts I've had? After all, our minds *are* more powerful computers than those we manufacture. Our culture simply hasn't developed a mental technology equal to our computer technology, that is, some set of techniques that would allow us to retrieve and process all the information stored in our minds.

I now remembered those times when I had touched a kind of Intelligence that was far greater than my own—specifically

those occasions when I somehow understood things without the usual mental study.* Associations were now flying through my mind: concepts of parallel processing within massive distributed computer networks combined themselves with concepts of lucid dreaming, telepathy, and various other extra-sensory phenomena that I'd read about. My own mind was a powerful computer; so too were the minds of all humanity—was it possible that we were already somehow connected, not physically but rather in *consciousness?* Was this the Intelligence that I had tuned into? If so, was there some way—some essentially *spiritual,* not just mental, technology—that would enable me to tune into it more deliberately?

If this were indeed the case it would certainly save me a whole lot of trouble! I wouldn't have to read everything that had ever been written—I could instead understand it all from the *inside*, from within the very minds that had produced those works! And that meant…

In one great leap of insight my mind suddenly arrived at a final conclusion: to know Truth I would have to tap into whatever consciousness pervaded everything and everyone in the universe—*past, present, and future; animate or inanimate.* For without tapping into all there was, is, and ever will be to know, without tapping into some kind of cosmic omniscient superconsciousness, I would never be sure that I really had Truth.

And what else could that consciousness be but the very Mind of God? Was it simply God that I sought?

* I'm talking of those experiences that I related in Chapters Nine and Ten, as well as the story of that presentation I told in Chapter Six where I had "tuned in" to a specific design process (page 81).

As if it were merely a technological reflection of what was going on in my mind, it just so happened at this time (the latter half of 1995) that the Internet or the "Information Superhighway," as it was then called, became all the rage. Everyone, it seemed, was suddenly talking about "getting connected."

To this point, Microsoft had been pacing itself with regard to the Internet as no clear direction had yet emerged from the industry frenzy. While there were specific teams within the company focusing on Internet-related technologies, such things were lower priorities among other product groups.

All that changed in early November. The media buzz had reached a fever pitch and industry analysts were asking questions. "Microsoft has been very quiet about the Internet thus far. What are they hiding? They must be doing something to establish leadership!" More and more they demanded that we share our story. Finally, Bill Gates promised to reveal Microsoft's plans. This would happen in one month's time at special conference on December 7th.

This was all well and good except for one small detail: we really didn't yet *have* a coherent story! We had plenty of applicable technology, of course, but so far as I could tell we lacked an overarching strategy that would bring it all together. With incredible speed and deliberation, then, we began to integrate everything we already had and invent whatever it was we didn't. Everything else was put on hold—the next month was 100% committed to The Internet Story.

It was immediately clear that OLE was a central part of our message: small software components were a natural and obvious way to add powerful capabilities to a web site. Indeed, the implications of component software in an environment like the Internet had been on our minds for years: with only a few minor refinements, if any, OLE-based components would be perfectly suited for use with the web. As a result, anyone and

everyone with OLE expertise—especially myself and the whole of OLE Program Management, which now included my old compatriot Charlie Kindel—was shanghaied into the project.

Thus I found myself suddenly yanked down from my lofty mental and spiritual skies to the hard reality of my responsibilities at Microsoft.

In many ways that month was really fun. In prior months we had been mired in hair-splitting minutiae; now, unable to afford such luxuries we just got to invent or re-invent as much as we wanted to! Idealism reigned as the immediate goal was to come up with stuff that at least *sounded* good on December 7th. Whether it was ultimately *practical* was beside the point—we'd have time to correct all that later.

At the same time, the unprecedented pressure of these few weeks was incredibly intense. I called it "Internet Hell Month." People were scared—the Internet's sudden importance and our relatively late entry into the game posed enormous threats to Microsoft's success and therefore to the value of people's stock options. In years past, the undercurrent of the company was a joyful aspiration to meet new challenges. Now it became **fear**. Everyone knew it; everyone could *feel* it, and it made life at Microsoft more miserable than I had ever known it to be. For this reason I count that month as the worst part of my entire Microsoft career.

Yet again, in retrospect, it turned out for me to be an opportunity that I wouldn't trade for the world.

You see, in the process of making the Internet the Most Important Thing in the Universe, Microsoft shifted its entire momentum. Energy was withdrawn from other commitments and redirected toward the Internet. Things that had been "essential" were no longer so; various long-term goals were thrown out altogether because they simply weren't going in this new direction. The world was ensnared in the web and Microsoft, as

a worldly company, had to follow.*

My dreams for the OLE technology were thus, at a minimum, put on indefinite hold. The Internet was, to me, a perfect delivery vehicle for software components—its role, in my mind, was merely supportive of OLE's broader purpose. But the situation was reversed so far as Microsoft and the whole industry were now concerned: OLE, already complete in itself, became merely a supporting technology for Microsoft's Internet strategy. Whatever real significance it had enjoyed on its own was now gone.

I was utterly heartbroken. Helplessly I watched OLE's tremendous potential recede into the technological distance until it finally disappeared over the horizon. And with it vanished that to which I had essentially dedicated my professional life. Not that I still didn't believe in my long-term vision, mind you—even to this day I believe it worthwhile. But without some kind of support within Microsoft I doubted whether I, as a solitary engineer, had the long-term energy and perseverance to make that vision a reality.

More doubts followed. Could component software alone help save the world from a destructive breakdown? I had to admit that the idea was somewhat far-fetched. Yes, the kind of computing environment I had envisioned would be a wonderful thing, but would it really affect mass consciousness? Would it really help people develop a higher sense of cooperation? Would it really be part of a cultural breakthrough? In all the history I had read it was obvious that "advances" in philosophy, sociology, science, technology, the arts, and almost every other field

---

* Considering the size of the company and the amount of energy that had to shift, it's astounding that Microsoft not only survived the initial threat but has continued to be a major player.

of human endeavor had rarely, if ever, been the direct cause of widespread spiritual awakening *unless they were first inspired by a high degree of spiritual awareness.* In addition, those advances required a certain degree of receptivity on the part of people in general—for without receptivity, spiritual growth simply isn't possible.

Thus where the cooperative element of component software was concerned, I couldn't fool myself into thinking that anything but a small percentage of people would be open to such a subtle influence. And as for my own level of spiritual awareness? Well, I couldn't even pretend to think that I came anywhere close to the likes of Krishna, Moses, Buddha, and Jesus, let alone many other great men and women of history. For all the good my envisioned plans might bring, I knew that I just didn't have what it would take to make them a reality. There was still much work to be done on myself.

After our presentations on December 7th I decided to take the rest of the year off (I still had a bunch of vacation time to use). Finding myself completely stripped of almost everything that had given any kind of direction to my life, I had nothing left but to wholly commit myself to the one thing that remained: my search for Truth. My goal was to work through all the books still on my reading list, hoping that these, especially—some of the most profound and thought-provoking I'd ever read—would give me some kind of guidance. After I read that one subtitled "The Search for that Something More," in fact, I wrote in my journal, "I'm close with this one... very close now."

Yes, I felt very close to the answers I so desperately sought. "Just a few more books," I told myself, "and I'll surely have it!" With redoubled zeal I continued my reading, now at a blinding pace of three or four books a day!

But as the pages wore on I felt more and more like I was

caught in Zeno's paradox, the one about how an arrow in flight can only reach its target by first traversing half the distance. Then it must travel half the remaining distance, then half of that remaining distance, and so on. But no matter how many times you halve the distance, you'll never reach zero. Therefore, so Zeno argued, it's impossible for any arrow to ever actually hit the target.

In my case, it seemed that no matter how much I read I kept coming up just short of the Truth I sought. Day by day, as my reading list dwindled to a mere handful of titles, I found myself—to change the metaphor—backing step by step into a very tight corner, finally to realize, like some hapless Greek peasant confounded by the local sage, that I was completely stuck. I had read so, so much, but I had not reached my goal—only exhaustion. What was I to do now? Was there any way out? Was there anywhere left to turn? Was there any resolution? With so much confusion, it seemed that a breakdown—a complete, personal breakdown—was inevitable…

Little did I know how close I was to a real breakthrough!

You may be wondering by now whether my wife Kristi was aware of my inner upheavals. To be honest, no—save those few times when my intensely inward search leaked out in some way. She, of course, witnessed my passion for reading books. But as I hadn't really been able to broach spiritual subjects with her in the past, conversations typically gravitated toward more mundane matters.

Kristi was nevertheless going through an inner process of her own. In the periods of solitude created by my frequent traveling for sometimes as long as three weeks, she was also searching for answers to her own deepest questions.

Outwardly this search expressed itself as increased stress, which ultimately took a toll on her body. Knowing that this wasn't particularly healthy, she sought ways to relax. One of her co-workers, a good friend with whom Kristi shared some of her frustrations, suggested she take up some kind of meditation practice. As a starting point her friend suggested a visit to a well-known metaphysical bookshop in Seattle.

Taking her advice, Kristi visited the store sometime in July or August of 1995 and bought a couple of titles that she found attractive. When she got home she found a little flyer inside the bag. It was an advertisement for a meditation class being offered by the yoga center affiliated with the bookshop. As "relaxation" was included among the listed benefits, she signed up for the class series in September, during which time I'd mostly be away giving lectures in Europe.

She was not disappointed. She found the meditation techniques marvelously effective and was even more inspired by the genuine joy of the instructors she met. She was so touched, in fact, that despite various mental objections she registered for a follow-up class. This, too, she found instructive and deeply inspiring.

She was also impressed that these classes were not filled with the kind of fluff one encounters in a typical New-Agey or self-help workshop—they were quite serious! Qualities like peace and joy and love were spoken of as aspects of God. Techniques of measured breathing and meditation were offered not so much as methods to *produce* inner peace and such, but as methods through which one learns to attune oneself to those qualities as they already exist, albeit latently, within your own Higher Self. Thus to truly find peace you must open yourself to Peace as a reality in itself. To find joy, open yourself to the Joy that pervades all creation. Anything else—such as the peace you find by ridding your life of some petty annoyance or the joy

found in the satisfaction of some desire—is only temporary. True permanence can be found only in that which is already Eternal.

It was clear to Kristi, then, that the "yoga" offered by this center was much more than just a system of bodily postures. It was the integration of one's small self with a larger Self (the real definition of the Sanskrit term). It really offered a transformative approach to every aspect of living, and one that was based not on belief or any institutional affiliation but on direct, individual experience.

This approach spoke to her and awoke an interest in spirituality that took me by surprise. Having been raised in a secular family, she had firmly stood apart from anything to do with God or variants thereof for almost the whole time I had known her. At the same time, she deeply believed in qualities like love, kindness, and generosity, and practiced them to a much greater extent than most of the so-called "religious" people I had known. My surprise, then, was a pleasant one. I was happy to see her finally delving into matters that had occupied my own thinking for many years. I was even happier to see her more full of joyful inspiration than ever before.

As you might expect, then, I was becoming very interested myself. In fact, starving for some real joy to counteract the intense fear permeating Microsoft, as well as the almost hopeless desperation I increasingly felt in my reading, I was ready to jump right in! Nevertheless, I stayed out of it for the time being. For one thing, I wanted to give Kristi the freedom to explore it all on her own terms. I also wanted keep a keenly skeptical eye on the whole business; my upbringing in a conservative Lutheran church combined with my own distaste for any kind of organized or formal spirituality made me a bit suspicious, especially of anything with an Eastern influence. After some weeks of observation, though, I saw absolutely no

cause for concern. Whatever Kristi was doing was having a profoundly positive effect on her life, the people she'd met were wholly sincere, and whatever they taught and practiced was wholly genuine, even if it was a bit out of the mainstream.

I finally ventured into this scene myself at a mid-December Christmas party held at a residential community north of Seattle where a number of the group's members live to support each other in their common spiritual goals. Kristi invited me along because others were eager to make my acquaintance, presumably because of the stories she'd told about me.

As we drove into the community apartment complex a definite sense of belonging welled up within me, that same feeling I'd had over seven years earlier when I walked into my first Microsoft interview: *I was home!* I couldn't explain it, but it was amazingly clear: I belonged with this community, whatever the heck it was; and all these people—ostensible strangers all—were family. Long into the evening we sang carols and talked together; Kristi and I were among the last to leave.

A few days later Kristi called me from work. She was thinking about taking a short trip over the New Year's holiday to the retreat center located at the group's primary community in northern California. As there was just enough space left for two people, she wondered if I might like to go. "Sure! Why not?" I had nothing better to do—I could certainly use some time away from home, away from reading, and away from all the heavy thoughts that pervaded my reality.

On December 28th, by which time I had become completely stuck in my philosophical corner, we arrived at the retreat and again…I was home. Relaxing almost instantly to a depth I'd not yet plumbed, I simply dropped our bags in our room and gave myself completely to the retreat's flow of yoga postures, meditation, instructive classes, and the uplifting company of like-minded souls.

As with my time at Yachats, I found so many worries and concerns melting away in the magnetic peace of the retreat. This time, however, instead of feeling lonely or empty I felt full of meaning and joy. That Presence of Love was fully alive in my heart, as if celebrating with delight.

For here I finally came into what I was looking for: a vision of life embracing everything from the lowest mundane needs to the highest spiritual aspirations, directing them all toward the realization of the full potential of *every* human being. It was a vision that included God, yes, but a God that dwelt within as one's own Higher Self, a God that one could *know*.

This vision was expressed clearly in a book called *The Essence of Self-Realization,* a collection of sayings by Paramhansa Yogananda, the widely-revered yoga master and author of the spiritual classic, *Autobiography of a Yogi.* In reading these passages I was amazed to find confirmation of what I considered to be even my more unorthodox ideas. For the first time I was truly given permission (from a recognized spiritual authority, that is) to relate to God in a manner that was meaningful to *me*. All my life I'd only heard sectarian variations on "Our way is the only way…" along with its companion thought, "God will punish you if you think otherwise…" Is it any wonder that I pushed it all away? Now, however, I could conceive of God in a way that I could truly embrace and love—as, indeed, I had already been doing. What's more, I discovered that all those "core needs" I had worked out a year earlier—wholeness, peace, truth, freedom, wisdom, and love—were not only contained in Yogananda's expansive vision of the Divine but wholly transcended by it. My "needs," in short, and what I truly sought, could be summarized in that one word: God. Not as anyone else defined it, but as all my sincere searching had enriched it.

I knew now that all my years of reading and searching were rapidly coming to a resolution. As I continued to read *Essence,* I

was pleased to rediscover many of the same bits and pieces of Truth that I'd marked on so many scattered pages of so many different books; I was even more delighted to find many other things that I probably would *never* have found on my own along with definite techniques of practice to make it *real.* And now here it was all right in front of me, all in one place, all brought together with simplicity and clarity!

I cannot say how profoundly grateful I was in that moment. While I'd been willing to try putting it all together myself, I was suddenly relieved of a tedious chore that would have probably consumed many uncertain decades to come. My and mind heart breathed huge sighs of relief.

Of course, now that I had finally found the Truth and the "spiritual technology" I'd been looking for all these years, it was pointless to simply go on reading about it all: *it was high time I actually started living it!*

It was on New Year's Eve 1995 that my life finally took this decisive turn. At the community-wide celebration that evening I had a chance to reflect on the people I'd met who had been personally, and as a community, living these truths for most of their adult lives. While engaged in outwardly intense activity—of the sort that would grind down most people into nervous wrecks—I yet saw generosity, vitality, joy, patience, considerateness, deep inspiration, and above all, a true centeredness and sincerity. It was a far cry from the worldly-minded people I knew, like a few of my Microsoft associates who seemed to endlessly whine about the value of their stock options.

I was also impressed by the degree of self-offering I saw in them. This was key. Whether one was a teacher, a carpenter, a lawyer, an artist, a doctor, an administrator, a mother, or a computer geek like myself, each shared an inner commitment that produced a deep radiance in their eyes. It came not from who they were, where they lived, or what outward role they

played but from the kind of consciousness they held in themselves and supported in one another.

Pondering these thoughts in the glory of a setting sun and reflecting on where my own life was going, everything finally came together. Whether the world or the software industry had come to a "critical juncture," I did not know. But *I* certainly had, and now I had to make a choice. Would I continue to chase only after worldly dreams or would I give myself into this greater reality that was literally staring me in the face? Everyone I met at the California community and their Seattle branch seemed to have what I most desperately wanted. They'd gone far to develop the spiritual qualities that I sought to develop in myself; some had even achieved great success in almost every one of my more mundane interests. As you might recall from Chapter Five, and as I was now recalling myself, I hoped one day to leave the computer industry to explore things like writing, music, and photography. With so many examples in front of me it became clear that anything I might do along those lines (or even anything I might still do with technology) would have little immediate or lasting value unless they too were inspired—and blessed—by my *own* self-offering to God!

I thus decided that if dedicating myself to the principles and practices of daily meditation, devotion, self-discipline, and service to others was the most effective way to fulfill both my outer *and* my inner goals, well then…sign me up!

God must have smiled in that moment. After working patiently on me for years, I had finally accepted his call and set heart, mind, and soul on the life path of deliberate spiritual living. I had felt trapped in a tight corner because I was looking in the wrong direction. I had only to turn around and lo! my life, my world, and my universe expanded outward to embrace Infinity.

CHAPTER FIFTEEN

Enoughonaire

"New car, no new car; new house, no new house." My friend and colleague, Nat Brown, often gave us this take on the volatility of Microsoft's stock price.

Nearly every Microsoft employee hung on the latest real-time market quote of the MSFT ticker symbol. You could usually tell whether the stock was up or down by the general mood in the corporate hallways: buoyantly cheerful or heavy and sour. Imagine the effect of a $10 move in either direction!

Our reactions were partly due to Microsoft's generous can't-lose Employee Stock Purchase Program (ESPP) into which employees were allowed to set aside up to 10% of their paycheck. At the end of each six month period those funds were used to buy company stock at a discount—specifically a 15% discount off either the stock price at the beginning or the end of the period, whichever was lower. You came out ahead, in other words, even if the stock tanked. No surprise that most of us set aside the full 10% throughout our entire careers.

ESPP shares were only a small part of our concern over Microsoft's stock price, however. By and large, the vast majority of employee interest came through what we called the "Golden Handcuffs": Microsoft stock options.

A stock option, in case you aren't familiar with the details, is the right to buy a share of stock at some future date for a predetermined "strike" price—usually the market price on the day the options are bought or issued. People purchase options on the speculation that the stock will go up by the time the option can be exercised. If they're right, they can then purchase shares at the lower price and realize an immediate and often substantial gain. So why not just buy the stock itself? One word: leverage. The cost of an option is usually well below the cost of an actual share. With the same amount of capital you can acquire many more options than outright shares while still realizing the same per-share gain, making options a simple way to get rich quick, so to speak. The downside is you can quickly go poor as well! If the market price is below the strike price when the options are set to expire, those options become absolutely and irrevocably worthless.

Microsoft was, at least until the year 2000, something of a legend in the stock market. My father still kicks himself for an investment he made in 1987 when he wanted to put $2,500 into a technology company with good growth potential. He whittled his choices down to a well-established computer manufacturer named Kaypro and this young upstart software house called Microsoft. Need I say more?*

With its track record in the market, Microsoft effectively used stock options over the years to motivate and reward its employees. Depending on your position, the relative importance

* Kaypro eventually went bankrupt. At Christmas some years ago, after I had achieved some success at Microsoft, my father gave me the now-worthless Kaypro stock certificate as a gift, framed up alongside the check with which he bought my first computer. I think he titled it something like "Study in Contrasting Investments," proof that the best investments are made in people rather than paper.

of your work, and your personal performance, you might have received a grant (that is, an outright gift) of stock options every six to twelve months. Those options, however, had no initial value for two reasons: one, the strike price was the same as the current stock price and two, the options were not yet "vested"—another way of saying "you can look but you can't touch." The first quarter of an option grant became available to exercise after eighteen months; another eighth vested every six months thereafter. This meant it took four-and-a-half years to gain the full benefit of any given grant.

Of course, by the time the vesting period was over the market price had usually gone up significantly and the stock had probably split a couple of times to boot. So a typical grant of 1,000 options with a strike price of $50 could easily become 16,000 options with a strike price of $3 and change. And with the stock itself generally hanging (in the 1990's, at least) between $70 and $150 a share…well, you do the math!

"Are you satisfied with your base salary?" asked one of the questions on the annual company survey. "Somewhat," was our perpetual reply. "Are you satisfied with your overall compensation?" asked the next. "Oh, *YES!*" No one could doubt that stock options were definitely "golden." They were a constant reminder for us to put forth our very best effort regardless of the challenges, setbacks, and long hours that were simply part and parcel of the Microsoft culture.

Of course, the stock option program wasn't designed merely to inspire employees to new heights of dedication, commitment, and occasional martyrdom: it was also designed to keep capable employees in the company for the long haul. That's the part about the handcuffs.

You see, upon leaving Microsoft you flat-out sacrificed any and all unvested options. Gone. Poof! And even those that were vested had to be exercised within ninety days or they too dis-

solved into the Great Void. What's more, we weren't allowed to exercise the options to acquire actual shares of stock; options had to be sold for cash. So if you had any inkling whatsoever to walk out—because of stress, boredom, or the simple desire for a change of pace—you couldn't possibly do so without peeking at your personal stock option spreadsheet and your next bite of financial candy. Upon doing so, you couldn't help but think, "Gee, if I just stick it out a couple more months and finish the next vesting cycle…and with the stock still rising…." Yep—a few more fortnights and you could easily pad your net worth with another quarter million dollars, if not a fresh option grant altogether!

Suffice it to say that the scheme totally worked. The promise of easy wealth, so close and so tangible, was difficult to give up on a whim or even under duress. And if you found a reason to stay on for one more vesting period you could probably find reason to hang out for a few more. After all, the system produced millionaires by the cartload. Why fight it? We donned the handcuffs quite cheerfully.*

Now let us pause for a moment and consider an ordinary middle-class person who was raised in an ordinary middle-class family—you know, the ones who fill in the "Annual Income $40,000-$70,000 per year" bubble on those corny surveys and who mostly dream of sending the kids to a decent college and having a reasonably comfortable retirement. Now plunk a few million dollars in their lap—or even just a few hundred thousand—and suddenly, like lottery winners, they lose any and all sense of proportion. What's blowing $50,000 on some flashy toy

* Due to the long bear market in tech stocks, Microsoft shareholders voted to end the program in 2003 and replace it with one based on actual shares. The Golden Handcuffs had finally lost their luster.

when it only means changing a number like $3,487,922 to $3,437,922?

What indeed? Around Microsoft, paying cash for a Ferrari, a Lamborghini, or even a million dollar house was not unheard of. Shelling out twenty grand for membership in an exclusive country-club hardly raised an eyebrow. Forking over four hundred bucks an hour for personal hands-on fighter-jet training was considered "quite cool." And going on extravagant and exotic vacations to places like Micronesia or Madagascar was almost a social expectation.

One of my associates, who will remain anonymous, was perhaps a quintessential example in this regard. Being a fifteen year veteran of the options game, he once went out and bought a 31-foot motor home—with cash—just because he "wanted to know what it was like to drive one." He wore a Rolex on the singular justification that it "kept good time." And he invariably upgraded his airline tickets on every business trip if only for the sheer privilege of asking the rest of us how we liked "cattle class." He even once considered signing up for a record-setting 23-hour round-the-world flight on the Concorde to the tune of $25,000. To his mind, it was perhaps a reasonably inexpensive way to earn a small place in history.

At least I derived some free amusement from his flamboyance—on a number of business trips together when we had nothing better to do than totter around whatever city we were visiting, I'd try to get him to spend as much as possible on something he absolutely didn't need. My top score was $2,300—we had paused outside a camera shop in Munich where I admiringly praised a beautiful Leica R4 as "one of the best SLR's ever built." He bought it on that principle alone. To his credit, I think he did actually use it…once!

All of this is to say that the whole stock options business engendered a certain direction to the development of one's

lifestyle. I say this from experience. Having received various option grants over the years and watching their value grow by leaps and bounds, my wife and I found ourselves more willing to express our abundance outwardly. First we bought a brand-new 2800 square-foot home in a fairly upscale suburb north of Seattle. Then we populated it with not just a brand-new car but with the latest appliances, stately teak bookshelves for my den, a big screen-TV and hi-fi VCR (no DVDs yet), an elegant set of bedroom furniture, and—something I had wanted for years—a very new and very black grand piano. In addition to all this, we were already planning for a good assortment of additional luxuries. There would be new furniture for the rest of the house. There would be the automatic sprinkler system for the front lawn and ornate landscaping for the back. There would be the fine lattice-covering for our large deck and another new car. And if Microsoft stock continued to cooperate we could not only hire some professional help to take care of it all, but perhaps start the whole act over again with some ocean-side retreat. Someday we might even be able to expand into the likes of Micronesia, Royal Coachman, and Rolex!

Yes, who could doubt that happiness was and would continue to be ours? We, like many other Microsoft families, had the means to immediately and rather painlessly fulfill each and every material desire, and were making a plucky attempt at doing so! It was the normal and even the expected thing to do for those in our position. Being surrounded by other such people on all sides, why should we ever stop to question? All our money and possessions were, as far as everyone was concerned, giving us happy, fulfilling lives, free from the cares and worries and normally plague the bulk of humanity.

But when we shifted the focus of our lives from worldly to spiritual aspirations we became exposed to another reality. In particular, we got to know a good number of people whose lives

were based on a fundamentally different assumption: they lived for God, not gold. Wealth, to them, meant inner joy, not possessions; fulfillment was defined in terms of desires transcended, not desires satisfied; and security was a matter of contentment and proven faith rather than diversified investments.

As a result, the direction of their lifestyles departed quite radically from that of the stock-option circus. For example, many folks at the spiritual community near Seattle (which we were now visiting regularly) were happily living in homes or apartments that were barely larger than the *hallways* of our house, and were nowhere near as new. Their cars were humble, their furniture functional, and any electronics they owned were usually of the basic sort. And while some people owned various kinds of decent musical instruments, there were certainly no grand pianos among them. Indeed, few could really afford such luxuries: in comparison to our combined six-digit income plus stock options, these individuals got along fine with about $20,000 a year.

According to all outward measures and standards, then, a good portion of this new spiritual family into which we had come essentially lived in so-called "poverty." Yet what they lacked in the material realms they more than made up in the spiritual. They were all so happy, even *joyful*—far more so than everyone else we knew! In their lives of meditation, service, and devotion, they simply didn't need luxuries: they were content with having *enough*. Many were also quite generous with what monies they did acquire, and they were equally content to let go of everything if it ever became necessary to do so.

In contrast, how free were we? When we took a long, honest look at our lifestyle, we came to the uncomfortable realization that its increasing opulence was far more binding than the "golden handcuffs" that made it all possible. It's one thing to have money; it's something altogether different to depend on it

as the very source of happiness. To do so only enslaves you to perpetual discontent, forcing you to seek your fulfillment in ever-more extravagant ways. We thus came to realize that this dependence, more than anything else, would keep us chained to the grindstone for a very long time, options or no options!

In our honest self-examination we also realized that our so-called riches weren't actually making us all that happy. As a matter of fact, a number of them seemed intent on making us downright miserable. I mentioned our 2800 square-foot house. Impressive, yes? The envy of our less fortunate friends, right? Well, little did I realize the kinds of demands it would make of me. The front lawn, for instance, installed as it was on top of solid clay, insisted on wilting no matter how much I pampered it. The Swedish finish wood floor—all 600 square feet of it—demanded (according to the instructions we were given) that we clean it every other week on our hands and knees. The air-circulation filters had to be hosed off every month. It took us a good hour-and-a-half to vacuum the endless oceans of carpet.

Then paint began peeling off the façade of our entry way. The back lawn succumbed to the guerilla forces of clover, dandelion, and chickweed. The lovely sun deck became parched and gray—demanding many hours of scrubbing, sanding, and re-staining. And the siding—which was discovered to be partly defective—brought with it the joys of a class-action lawsuit.

Need I go on? Need I mention dusting? Need I mention washing a dozen windows 15 to 20 feet above ground? Need I mention the wonderful capacity of kitchen tile grout to absorb beet juice? And all this was just the beginning: our new car, our fancy furniture—and yes, that big, black, grand piano—all made respectable contributions of their own to the increasing *complexity* of our lives. In fact, the more we looked the more we realized just how enslaved we truly were: possessed by our possessions. Wasn't wealth supposed to simplify matters?

Now let us recall for a moment the state of affairs at this point in this whole story. I had just been through a month at Microsoft that was, by my reckoning, sheer hell. My hopes and dreams for the OLE technology had been usurped by the Internet frenzy. What's more, my work in OLE Program Management—which I had entered with high hopes of helping the technology fulfill its great potential—turned out to be little more than playing mediator between groups of people who generally nursed a clear distaste for compromise.

My professional aspirations, in short, were pretty much in the gutter. At the same time, my spiritual aspirations were soaring higher and higher, as if to compensate for my technological disillusionment. Expansive vistas were opening before my eyes: new ways to live, new ways to serve, new ways to love. This, I knew, was where my future lay.

At the dawn of 1996, then, I was wholly ready to just walk away from it all: away from my career, away from all our possessions, and away from all the expectations that I'd been surrounded with my whole life. I even hoped that God would expedite the process—perhaps when Kristi and I returned from our retreat in California we'd find that our house had burned to the ground and that a pink slip awaited me at Microsoft! But rats! No such luck. The house was still there with all its stuff and I still had my job, same as ever. So I had to accept that however things were going to resolve themselves would happen one step at a time, with our conscious cooperation. God never forces you down a path…he simply invites you to take each step by your own free will. In our case this meant consciously choosing to re-create our entire lifestyle to more clearly express our new ideals and priorities.

This is actually what true simplicity or "simple living" is all about. Just as many people believe that possessions and wealth defines the "good life," others believe that simplicity means a

complete rejection of technology and material comforts. To live simply, they say, one should follow Thoreau and head off to some cabin in the woods, or, if that's too extreme, to get all into things like solar power, composting toilets, self-sustaining organic farms, and home-spun textiles, preferably in some remote rural area well away from the wretchedness of big cities. The truth, however, is that merely changing the outer forms of one's lifestyle will not automatically bring joy, happiness, or inner peace any more than wealth. In fact, "simple country living" can be as much if not *more* demanding than life in a metropolis. If you don't believe it, arrange to operate a rural farm on your own for a couple of months, or simply try growing, harvesting, threshing, and grinding your own wheat for bread! Then ask yourself, "Is this how I really want to spend the precious hours of my life?" Thoreau went to Walden Pond not to live in the woods but to live, as he wrote, *deliberately*.

Simplicity means first being clear about your priorities—that is, knowing what you really want to *experience* in life.* It doesn't matter whether your priorities match up with anyone else's: what matters is that *you* are clear about what *you* want. With this clarity you can then focus all of your life's energies in that singular unified direction and surround yourself with an appropriate environment—what you own, where you live and work, who you associate with, and even how you think. This focus becomes the yardstick against which you measure every decision. Does this or that choice serve the fulfillment of your priorities? If it does, then it's a valid choice regardless of all other considerations; if it doesn't, then it's nothing but a waste of time and energy for you no matter how many good reasons there are to the contrary.

---

* A subject I'm taking up more fully in another book, *Finding Focus*.

Wealth is not a matter of money: it's simply having what you want. Simplicity is not a matter of any outward form: it's wanting only what you truly need and truly needing only that which is in line with your life's priorities. So many people feel enslaved by their jobs; in reality, one's desires, habits, and attachments are the real taskmasters. It is to serve *them* that our jobs, and the income from those jobs, seem so necessary; when desires are expanding, one's income never seems enough.

Again, the more you can define your true needs in terms of inner experiences—joy, love, peace, wisdom, etc.—the less you need depend on specific forms of fulfillment. While that exotic vacation, that new car, that fancy house, and that big, black grand piano might offer some fulfillment, there are probably hundreds of equally effective and far more economical means to the same end. Sure, they might not impress others to the same extent, but they also won't keep you bound to the grindstone—or stock option vesting cycles—for years to come! In simplicity and contentment there is real freedom.

As you already know from the Prologue, this shift in our life's priorities led me within a year to part ways with the world's greatest software company. I'll save that story for the next chapter; here it remains for me to explain how my wife and I parted ways with everything else.

First came a basic change in our attitude toward money. For the longest time, the numbers in my stock option spreadsheet (which I seldom failed to update with the latest price quote) were just numbers; all I knew was that they represented a pretty big pile of dough. But as I was now beginning to contemplate leaving Microsoft, I had to look at it all with a new perspective: what would I actually *do* with it? What did it *mean* to have stock options worth a half-million dollars or more? What, indeed, is this mysterious thing we call "money" good for, anyway?

Then I once again remembered that long-forgotten dream. Years ago (as you will again recall from Chapter Five) I had planned to someday leave the high-tech industry for more artistic, scholarly, or spiritual pursuits. Back then I wanted to save two or three years' worth of living expenses to support the change—sufficient money was just a tool to make it possible. Now, even with a big house and its matching mortgage, I found I had enough for a whole decade! I already had, in other words, what I *needed* to have: anything beyond it was just an added bonus.

With this in mind, and with my career in something of a lull, it seemed prudent to begin diversifying our resources into something a little less volatile than Microsoft stock. This was a huge step. By forsaking the promise of future gains, it was an affirmation that I already had enough and didn't need to wait any longer to start making changes. It was the first yet most crucial step off the treadmill.

I started by exercising about ten percent of my options in early February 1996 without even knowing where I would invest the proceeds. But no sooner did the exercise go through than God presented us with a unique opportunity: we learned that the community where many of our new friends lived—and situated only a few miles from our house—was in need of new investors. Inspired by the ideals upon which the community was founded, we both felt to offer our financial support.

Now when we made this initial investment we weren't at all thinking of moving out of our house: we just wanted to help. But God had other things in mind and responded quickly to our openness! In particular, we both began to feel that we should do some estate planning. Our assets had grown considerably in the past year alone and with the house, the new car, the piano, and everything else, common sense dictated that we assemble our wills.

For a while we were wholly occupied with nothing more than the testamental legalities. But you can't work through the process of writing your will without thinking about that little thing called death. Then it all becomes very personal.

"What would you do," Kristi asked me one evening in April, "if I suddenly died?"

"Well," I replied, "the first thing would be to move into the community, so I'd be surrounded by supportive friends."

We paused for a moment, deep in our respective thoughts.

"And what would you do," I continued, "if I died?"

"Well," she said, "I'd move into the community myself!"

We looked at each other with a smile then burst out laughing. "So why are we waiting for one of us to die? Why don't we just move in together?"

Shortly thereafter the unit to which we'd felt most attracted opened up. I had specifically noticed it when we first drove into the community back in December. Looking up into its warmly-lit living room, I remember thinking to myself, "That's home." The only problem was that we couldn't possibly fit everything from our house into a mere 800 square feet! A bunch of it would simply have to go.

Under normal circumstances, shedding the majority of one's possessions is a painful—if not traumatic—experience, even if those things have proven themselves a complete burden. But for us it was nothing short of joyful. Honestly! We found great inspiration in a wonderful book called *Your Money or Your Life*,* which outlines a program through which anyone of any means can achieve financial independence, with or without stock options. Using this book as our guide, we spent the next

* By Joe Dominguez and Vicki Robin; this book was instrumental in changing our whole consciousness about money and possessions.

few months gleefully exploring how little we could really live with. The key was learning to see everything we owned as an investment of life energy. To own a thing means to freeze some portion of your life into a static object—so that thing had better be worth the investment! I also realized that everything we owned had a very real cost of ownership (needing somewhere to put it), and also defined, in a very real way, our future. Owning a thing presupposes that you'll eventually *use* it. So imagine yourself actually using that thing and ask whether such activity fits with your life's priorities. If not, let it go!

Thus we found it easy to throw out all kinds of dead-weight, like the boxes of my old schoolwork dating back to kindergarten. We also took part in our neighborhood's mega-garage sale, letting go all kinds of stuff that our future didn't need, like board games, extra stuffed animals, old records, and aging electronics. For the larger pieces—like most of our virtually new furniture and the big-screen TV—I took advantage of the semi-classified ads in the Microsoft company newsletter. Whatever remained I dropped off at the doors of Goodwill, the Salvation Army, and St. Vincent de Paul's thrift shop.

Finally it came down to the last two big-ticket items: my precious grand piano and the house itself. To be honest, I didn't really want to relinquish the piano. Secretly I believed that it could somehow fit into an apartment and not be a nuisance to our neighbors. I even thought it might contribute to our overall simplicity—by taking up the entire living room it would eliminate the need for any other furniture!

Well, let's just say that reason prevailed. If I really wanted a piano there were plenty of electronic alternatives that would take up far less space, require little or no polishing, need no routine (and expensive) tuning, and include—much to the pleasure of our neighbors—a headphone jack. So with a little sigh, I gave it up.

Then I had a good laugh at myself—I suddenly realized that the piano had been the sole reason for my even wanting a house in the first place! I remembered how I had once gotten upset at Kristi for not—in my mind—trying hard enough to find a job after she got her Master's degree in late 1992. We had agreed that we would start looking for a home as soon as we knew what we could afford. But it wasn't the lack of a house that upset me—it was the fact that not having a house kept me from having a piano!

I laughed even harder when I clearly saw how many other so-called "necessities" had come from that one desire: the lawn-care equipment, the furniture, the extension-arm dusters, the bookshelves, the fireplace insert, the new towels to match the bathroom colors—you name it! All for the piano!

After a laborious process we finally sold the piano and the house—both, I might add, at significant losses. But in my mind the lessons we had learned were far more valuable. God had taught us what it truly meant to own a thing and the trouble you can get into (debt or no debt) with even the most innocent of material desires.

With fitting serendipity we moved into the community on July 4th, 1996, declaring our independence from excessive possessions, from discontentment, and from the fleeting fulfillments of the "good life." No longer would we feel compelled to seek happiness in things. No longer would we be mindless disciples of the Microsoft Lifestyle. No longer would we be bound by Microsoft Stock Options. We were free to dedicate our life energies in service to our highest aspirations.

And when I eventually cashed out of Microsoft in October and put the proceeds in various income-bearing investments, we also found ourselves in the position of financial independence: the interest income was more than enough to cover the reduced expenses of our downsized lifestyle—not just for five or

ten years, but for decades to come. Our focused simplicity had given us both a spiritual and financial freedom that we'd never imagined.

It had melted the handcuffs, leaving us only with the gold.

CHAPTER SIXTEEN

Fade to Light

As corporations go, and despite whatever faults or shortcomings it might have, Microsoft has been and remains one of the best companies in the world to work for. Microsoft offers leadership, for one thing: almost every project has some kind of influence, whether real or potential, on the overall direction of the global computer industry. This alone is enough to draw many of the most talented engineers and managers into the Microsoft family, making for a very dynamic workplace. And even if an overly ambitious project turns out to be completely impractical, there is yet great delight in doing your best to contribute to the world's technological progress. Sometimes the journey itself is more important than the destination; certainly much is learned along the way.

The money one earns on salary alone is decent; other compensations can make it exceptional. On top of that, Microsoft has one of the most comprehensive, generous, and open benefits policies around, providing medical, dental, and vision plus coverage for disabilities, mental health, and life insurance. It even subsidizes membership in local health clubs.

The cafeterias at Microsoft's corporate campus are great in size, number, and sheer culinary variety. For breakfast, lunch,

or dinner, Microsoft boasts the largest dining facilities in the entire Seattle metropolitan area as well as some of the lowest prices—thanks again to subsidies. Coffee, tea, soft drinks, and fruit juices are always free, ubiquitously available in the kitchenettes found on every floor of every building. For those putting in extra hours during "crunch mode" there are often free dinners and late-night snacks as well. And occasionally, for no good reason whatsoever, someone might come down the halls handing out Dove Ice Cream Bars.

Microsoft strives, by and large, to make its employees comfortable. The buildings at its headquarters are designed to offer a wall of windows in nearly half of the offices. And they are *real* offices—with doors and individual lighting. Sure, you might at times have to share an office with another person, but it's a far cry from the typical cubicle jungle and a blazing canopy of fluorescent lights.

Other perks include free and frequent shuttle service between each building and between different Microsoft sites in the greater Seattle area. There are several full-sized athletic fields at corporate campus for softball and soccer, as well as full-sized volleyball sands and basketball courts. Employees are also given $1,000 of purchasing power per year at the Company Store where one can buy any Microsoft product for only a few dollars above cost. Most versions of Windows, for example, run $20 to $30; Microsoft Office, $50-$75; any game or CD-ROM title, $10. Needless to say, a thousand dollars will easily get you two to three shopping-carts of stuff and an easy way to impress your friends and relatives at Christmastime.

In general, the working hours at Microsoft are very flexible—workdays start anywhere between 4am and 2pm and end anywhere between, well, 2pm and 4am! As described earlier in Chapter Two, the ubiquitous use of email and other team-management tools render it fairly unnecessary for everyone to be in

their offices at the same time. What matters is getting your work done: *how* it happens and *when* it happens are, for the most part, irrelevant.

Microsoft does its part to address traffic concerns, especially as the company continues to expand. Besides flextime hours, it's a simple matter to telecommute. Microsoft also subsidizes bus passes and vanpools to ease the strain, and if the demands of your work cause you to miss your normal ride, Microsoft covers the cost of a taxi.

The dress code is relaxed—hard to find a suit anywhere, even in the legal department! As once reported in *Micronews,* for example, there was only one man on the entire Windows 95 development team—Raymond Chen is his name—who went so far as to wear a button-up shirt and a tie every day. After much good-natured bantering, the seventy or so other developers on the team finally convinced him to "go casual" for once. In exchange, everyone else agreed to follow suit, as it were. Thus was born "Dress Like Raymond Day"—and yes, some actually had to go out and *buy* a shirt and tie for the occasion![*]

Microsoft's relaxed attire is due, in part, to the absolute abundance of free Microsoft logo-wear, especially project shirts and custom conference apparel. And these handouts aren't junk! On several occasions I became the proud owner of a fine denim shirt with tastefully embroidered insignia. In Developer Relations I even got a high-quality wool coat, customized with

---

[*] Many years ago, when Bill Gates was going to his first meeting with IBM about MS-DOS, he realized that he might not get in the front door without a tie. So he bought one on his way from the airport. When the meetings were over, Bill left it pinned to the bulletin board of the office that IBM had set aside for visiting Microsoft engineers. For several years thereafter, each visiting male engineer would don what became known as "*The* Microsoft Tie" for his meetings, then leave it pinned up for the next person.

my email name.*

Then there are all the other free goodies—besides those Dove Bars…yum!—that were given out from time to time. Sometimes they were relatively cheap trinkets like sunglasses, Slinky toys, yo-yos, and Frisbees. But more often than not we got really good stuff: walkmans, beach chairs, briefcases, clocks, binoculars, golf umbrellas, lunch coolers, watches, you name it. Sure, it was an underhanded way to make every one of us a walking Microsoft advertisement, but we were more than happy to do everything in our power to boost our collective pride…if not the value of our stock options.

If all this wasn't enough, the very quality of the company's employees is an added benefit—Microsoft's influence attracts people with ambition and energy; its sacred initiation ceremony—The Microsoft Interview—finds the very best among them. Personal commitments to Microsoft and its vision run deep. Typically your office is within only a few doors of someone who has been both critical to Microsoft's success and has made significant contributions to the computer industry as a whole. A Microsoft recruiting brochure once put it this way: "If you want to know something about MS-DOS or Microsoft Word, just walk down the hall: the people who wrote it are probably there!"

Even so, it was your present focus and effort that mattered:

---

* These jackets were given to everyone in Developer Relations in late 1991. On the back was a large red-and-white target with an arrow sticking in it to emphasize the one-pointed goal of our evangelism efforts: Windows. We loved our coats and wore them with great pride—except one night during Windows World '92 in Chicago. Everyone from DRG had a little get-together at a jazz bar a mile or so from our hotel. Late in the evening about ten of us decided to walk back. About halfway we suddenly realized that we were walking the streets of downtown Chicago, at midnight, with these big targets on our backs! Prudence hailed the next available cabs.

I can't remember a single instance when someone held up his or her past accomplishments in some gesture of superiority. In fact, I found that the greatest geniuses in the company were the same ones who always took the time to help you understand the most complicated technologies. Your very interest in their work was the greatest gift you could give.

Suffice it to say that Microsoft is a fabulous place to work.

It is also a difficult one to leave behind.

Nevertheless, through 1995 and on into 1996, many longtime Microsoft employees were calling it a career. My old friend Bob Taniguchi, for example, had left just after Windows 95 shipped in August of that year. Every week or so it seemed that some other veteran employee announced his or her retirement.

As I alluded to in Chapter Fourteen, Microsoft was going through a tremendous internal change—not so much in terms of organizational structure and such, but *energetically*. In years past, Microsoft had always been the underdog. When it got into the PC operating systems market with MS-DOS, Microsoft was a speck of dust compared with the likes of IBM. When it got into the applications business, it couldn't shake a stick at Lotus and WordPerfect. And when it got into graphical user interfaces with Windows, the Apple Macintosh ruled the realm.

Under these conditions, the general attitude in the company was looking upwards. We had peaks to climb. We had markets to win. We had technological innovations to prove. With these aspirations came an expansive spirit that welcomed every new challenge with creativity, joy, and outright gusto. That spirit, in turn, produced a wonderful unity within Microsoft. Those who were naturally competitive and inspired by the struggle to "gain territory," so to speak, worked harmoniously with less competitive sorts, and I count myself in this latter group, who were inspired by the potentials themselves that we were trying to realize.

Then Microsoft won. By the end of 1995 Windows and MS-DOS were running on 90% of the world's personal computers; Microsoft Office on 98%. In just about every arena, Microsoft was the undisputed leader. Microsoft was King of the Hill. This dramatically changed the conditions: those who were motivated by the struggle to gain territory found themselves with the new task of *defending* what we'd won. In contrast, those who were motivated primarily by the upward climb itself suddenly found themselves with nowhere to go, with nowhere to look but *down*. And when we looked down we saw everyone *else* coming uphill to knock us off. Indeed, some were charging uphill with a vengeance! Intensely jealous of our success, or vowed to get even for some past offense, everyone seemed out to get us.

Under such novel pressure the underlying consciousness at Microsoft shifted. Microsoft wasn't about to surrender: it had every right to defend what had been won. This defensive posture thus became the predominant spirit, full of fear, anxiety, and disharmony. More and more of our energies now went toward merely preventing others from pulling us down; less and less remained for climbing any higher than we had already come. Infighting increased, and that joyful, upwardly aspiring growth we'd enjoyed for so long all but disappeared. And without that growth many of us lost the deepest motivation behind the intense effort we had once been able to sustain.*

Exacerbating the situation even further was the company-wide shift to embrace the Internet. With so many veterans jumping ship, Microsoft had to quickly recruit a fresh corps of soldiers who were thrown into battle from day one, ready to

---

* Fortunately, it seems that Microsoft has come out of this defensive posturing in more recent years with the launch of various new and expansive initiatives. Some who had left years ago (like Bob) have since rejoined the company to work on some of these new projects.

sacrifice all in this new campaign. And like the thousands of youths who were once shipped off to Vietnam, it was all they could do to fight for their lives. As a result, the undercurrent of fear and anxiety became even stronger. Infighting increased further, and mistakes that would never have happened in the past now arose with alarming regularity. Those of us who had been around during the Apple lawsuit a few years earlier, for example (see Chapter Twelve, pages 158-159), had learned to never say anything suggestive of an "anti-competitive" nature in any internal communication, especially emails, because it could be resurrected during a legal discovery process and used against us. Most new hires, on the other hand, didn't know any better. In the emotionally-charged atmosphere of the time it was perfectly natural to talk about "crushing" our competitors and so forth when all it really meant was "create a product that would win in the marketplace." Taken at face value, these were the sorts of messages that Microsoft's enemies along with the U.S. Department of Justice and other governments later used to support their claim of Microsoft's monopolistic intentions.

So like I said, Microsoft was changing and people were leaving. In my own case, as you already know having come with me thus far, I was wholly ready to drop out myself. At the same time, I also understood that this really wasn't for *me* to decide on my own: in seeking to re-orient my life in a spiritual direction, I wanted only what God wanted for me. This meant attuning myself to a greater Will that knew how to harmoniously guide my life much better than I!

I also knew that I didn't want to leave in anger and frustration—Microsoft had been such an important part of my life for so long that it deserved another chance. What's more, many people both inside and outside the company were still dependent on my OLE expertise. I didn't want to just leave them out in the cold! I also had to admit that while the Internet had

pretty much shot down my high-flying dreams of World Peace Through OLE, the technology itself wasn't dead. It still had a great deal of potential and there was still much I could do for it. Did God have another purpose in mind here besides the one I'd thought up? It was certainly possible. And, of course, there were all those great Microsoft perks that I'd have to give up if I left, especially a pretty good pile of yet-unvested stock options. Did God perhaps have some use for these? I just couldn't tell.

What I needed more than anything was clarity—in my own heart and mind above all. While still in California over the '95-'96 New Year's holiday, I took the opportunity to chat with one of the spiritual counselors on staff at the retreat, a deeply calm and joyful fellow named Wayne Palmer. We talked for a good hour and a half…well, that's not quite true: I talked to Wayne, Wayne talked to God. That is, while I blathered on endlessly about my "problems," Wayne simply listened, inwardly praying the whole time that I might discover the answers within my own self. Not the kind of "counseling" one normally expects, but I couldn't argue with the results! Though Wayne hardly said more than a few dozen words and never once told me what I should do or how I should approach my circumstances, I intuitively knew, at the end of that hour and a half, exactly what needed to happen next.

For starters, I knew it wasn't right to just run away from a painful situation, nor was it right for me to just be a doormat. If I was going to stay at Microsoft and not leave my teammates in the lurch, then I at least wanted a chance to work on those things I found meaningful and interesting. Otherwise, I had no real financial compulsion to remain: as mentioned in Chapter Fifteen, I already had enough assets to support my lifestyle for a good decade.

When I returned to Microsoft in early January, I sat down with my manager and let him know how I felt. To my relief, he

completely understood what I was going through and agreed to support whatever special projects I wanted to work on as long as I fulfilled the basic responsibilities of my position. For the next couple of months, then, I was able to relax into a more comfortable routine and concentrate on my work. My frustrations over the whole Internet business subsided and I no longer felt inclined to run away. In fact, I began to enjoy exploring the everyday applicability of a few of the spiritual principles I had so recently learned, setting my heart at ease that those teachings and one's career could come together in a beautiful harmony. The story told in Chapter Thirteen, for instance, took place during this time.

Meanwhile, an interesting thing happened to my public image as the "OLE guru." While my personal life had undergone drastic changes, people both inside and outside of Microsoft still saw me as the same ol' guy and still depended on my expertise. The dozens of emails I got every week asking for advice were certain proof of this! Could I possibly shed this responsibility without causing a great deal of pain and inconvenience to others? So long as people needed me I almost had a sacred duty to stay at Microsoft and play the role.

Well, God seemed willing to bail me out of this one: within a matter of weeks I was relieved of both the responsibility and the title. Part of Microsoft's whole Internet strategy, you see, was to come up with new names for old technologies in order to re-energize them. For example, the software components that OLE made possible were, before this time, called any number of uninspiring names like "OLE Compound Documents." Boring! This just wasn't going to cut it in the New Economy. So everything got renamed with some derivative of the word "Active" or, better yet, "ActiveX." "OLE Compound Documents" became "Active Documents"; what were known as <yawn> "OLE Controls" became <dude!> "*ActiveX Controls*"; and so on.

The result of all this was that I, personally, became entirely disassociated from Microsoft's so-called "new" Internet technologies, even though they were the same as before. While I had been universally known for years as the "OLE guy," I didn't automatically become the "ActiveX guy" even though I'd personally assembled some of the ActiveX specifications.

This became abundantly clear at the next big Microsoft convention, the Internet Developer's Conference held in March 1996 at the Moscone Center in San Francisco. Once it was virtually a matter of course that I would be called upon to give at least three or four lectures at such an event. Here, however, I wasn't asked to do *anything!* I wasn't asked to give a talk, I wasn't asked to sign books, I wasn't even asked to staff the hands-on lab we set up. I only went to the show because everyone else in my workgroup was going.

Aaah! What freedom I enjoyed! Without having to shoulder the tremendous responsibility of being the "expert," I was actually able to walk around the show entirely unmolested for a change. I smiled gleefully as I remembered how people used to crowd around me so much that I couldn't go to the bathroom. Now they were crowding around the new gurus, the ones who were getting their first taste of name and fame. Indeed, I was so well ignored that nobody but a handful of sincere technophiles who caught up with me at lunch one day even seemed to know who I was.

Many people, I think, would be devastated by such a free-fall plunge in popular opinion. But I found it blissfully liberating—I'd finally become what I'd always sought to be: obsolete! No one outside Microsoft seemed to need me any longer.

Nor did anyone else inside the company need me, for that matter. By the end of March the daily flood of email asking me to solve some OLE-related problem was down to a trickle. This allowed me to spend the next month or two in blessed obscur-

ity, working on my own projects with no hassles and virtually no interruptions. Indeed, my real responsibilities in OLE Program Management were shrinking week by week—there was very little need for me to interact with my teammates on much of anything.

While sitting in this state of professional relaxation, if you will (to which was added our simultaneous downsizing on the home front, as told in Chapter Fifteen, pp. 225-227), I realized I'd been given a window of opportunity in which I had complete freedom to choose my future. I could easily write a third edition of *Inside OLE* and expand my expertise into the new OLE-related technologies that were in the works. These would certainly keep me busy. Or I could simply leave Microsoft without any complications whatsoever.

Each choice, however, entailed a very significant sacrifice. Choosing to stay meant another two or three year commitment to make it all worthwhile, thereby delaying my ability to give myself wholeheartedly to my new direction in life. Choosing to leave, as I've said before, meant sacrificing a bunch of unvested stock options, not to mention all those other benefits. What was the right thing for me to do? Was I actually *supposed* to stay? Again, its perfectly possible—indeed helpful, as this book has shown—to actively walk the spiritual path in the context of a corporation, as I'd already been doing for years without really knowing it. And maybe I was supposed to stay so I would have the means to support any number of worthwhile causes, as many other past and present Microsoft employees from Bill on down have done. Or was it time to expand my interests beyond technology as I had originally thought to do many years earlier? I just didn't know.

This was all very much like the mental gyrations I'd experienced when Microsoft had offered me that first co-op job, only now the stakes were somewhat higher! Fortunately, I knew by

now that the real solution was to step back from it all and just "give it all to God." This is another way of saying that since I didn't have the clarity to decide on my own, I first offered up the problems in prayer and meditation then tried to see what was really trying to happen. If I kept my heart and mind open, the circumstances that Life put before me would show the way.

And believe me, they did! In June I was *physically* moved out of the OLE Program Management team, matching the fact, it seems, that I had already withdrawn energetically. Thanks to the eternal Microsoft Shuffle my headcount in that team was opened up for a replacement. I was moved under a different management chain altogether and relocated to another office. Ostensibly I was still working somewhere under the umbrella of Windows NT, but you couldn't tell by looking. My office, in fact, was nowhere near the center of any meaningful activity (save the cafeteria, two doors down) and those working around me seemed no less adrift than myself. Nor did I really know *what* I was supposed to be working on. Though I continued to do a little bit here and there on my own projects, I was mostly just killing time…to be open to possibilities but also to finish the current vesting cycle of my stock options!

Of course, things couldn't possibly remain like this for very long—sooner or later someone would ask why I was still getting paid for doing virtually nothing. I think that God could also see that this sort of useless stagnation wasn't good for me either: lethargy and indecision are not conducive to spiritual growth. Not surprisingly, then, circumstances compelled me to make the most important choice of my career: whether, that is, to end it.

On a Friday afternoon in early August, the person who was apparently now my manager, a newly hired chap whom I will call Ned, asked to see me. As we sat in the cafeteria he told me that he and his higher-ups—the ones who were paying my

salary out of their departmental budget—no longer supported "whatever the hell you're doing." Instead, he wanted me to write what he described as "persuasive competitive literature" to combat the Internet efforts of companies like Sun Microsystems and Netscape Communications. This basically meant badmouthing them while primping up Microsoft's initiatives as some kind of Great Dispensation.

Blecch. I couldn't imagine anything more personally demeaning. I had always wanted to share joy and to help others find joy within themselves—it was simply against my nature to condemn others in the way Ned was asking. Moreover, I had always aspired to serve some higher purpose in my work, not just the interests of Microsoft's bottom line, its public image, or even my own bank accounts. This was especially true now that my higher purpose was set on experiencing God more and more directly.

I wasn't in any way upset by this, however. Quite the contrary! Inwardly I was laughing my head off. I had been on the brink of leaving Microsoft for a while, simply waiting for some kind of definite guidance. Ned didn't realize that he had just become, through his proposal, an unknowing messenger for the Divine Will!

Repressing a bemused smile, I rejected his offer. "Sorry," I said, "I don't do that kind of stuff."

Ned was put off, I could tell. He was one of the Not Yet Vested and was hoping that I would use my skills to help gild his newly donned Golden Handcuffs. As giving me the boot for insubordination wasn't a helpful option in this regard, he thus proceeded to try other forms of motivation.

"But you'll be so influential!" he promised.

"Nope, sorry," I quietly replied.

"You'll get the attention of all the important people in the company—and the industry!"

That was the last thing I cared about. I had experienced all the notoriety I could ever want.

"You'll get bigger bonuses! More stock options!"

I had enough already, why would I kill my soul for something I didn't need?

"You'll be powerful, famous, glorified! You'll be…"

Stopping him before he wandered beyond the realms of sanity, I was barely able to contain the swells of my inner mirth. By having Ned make such a hilariously absurd offer, replete with almost every possible temptation of power and glory, God was making my choice an obvious one. It was time for me to leave. But to not slap Ned in the face, I told him I'd think about it and let him know in a few weeks. (Killing time, dude, killing time!)

Other circumstances of Cosmic Coincidence reinforced this choice further. Even little things that people said in passing seemed to answer certain questions and offered new ideas for how I could focus my outward energies to support my inner life. Still, it was also apparent that no one else was going to take responsibility for this decision: it was something for which I had to find my own inner conviction. It would take great energy and determination to so resolutely walk away from a long, successful career—and all those unvested stock options. It would also take great energy, steadfastness, and devotion to walk the spiritual path while also recommitting myself to Microsoft if I chose to stay.

To strengthen myself for this choice, I took another week of retreat in California where I could relax from the details and renew my energies for the road ahead. After many hours of meditation and prayer—mostly prayer!—I finally realized that everything simply boiled down to a single consideration: what did I want my life to look like? Did I want to continue with technology? Did I want more worldly success? Or did I want to

now expand my experience and venture into unknown waters beyond my present self-identity? There would be many rewards and fulfillments either way…but which to emphasize?

Again, in my experience God never actually forces us down any particular path. He always leaves us with the option to refuse his invitations. And even if we exercise that option, he often comes back again later with another possibility. In this eternal love he gives us endless opportunities to grow. Whether we move forward, backward, or simply sideways is always left for each one of us to decide for ourselves.

In the end I decided that while giving up much potential wealth and professional advancement would be a loss, it was nothing compared with losing the opportunity I had at this point to give myself completely to the path of inner growth. If God wanted to drop more money and success in my lap by some other means, fine, but given that I had no financial or professional compulsion to stay at Microsoft there was little reason to make any degree of compromise. Just as I had once moved beyond the sphere of Product Support, it was now time to move beyond Microsoft.

Let me again make it clear that I did not consider my expanding spiritual life at odds with a continued career at Microsoft. Throughout my final year in the company I had many opportunities to integrate the two—as again exemplified by the story of Chapter Thirteen and also with the story of that last-minute presentation related in Chapter Six—and I know there would have been plenty more had I stayed on. But the real opportunity here was to greatly *broaden* that experience. You see, since leaving Microsoft (as you can read in the short biography at the back of this book), I've been able to explore this integration within many fields besides the computer industry. These include education, construction, forest management, and retail sales, to name a few. With these diverse experiences

I'm much more able to help people in all walks of life find new and creative solutions to their everyday challenges. I don't think I'd be able to do this as effectively had my experience remained solely confined to Microsoft.

When I returned to work after my retreat, I sent a piece of email to Ned—as well as everyone I'd ever known and worked with—to announce my retirement. I set my final day in early November, giving me the necessary time to tie up a few loose ends, fulfill a couple of conference commitments, blow off a couple weeks' worth of vacation and sick days, and, of course, vest a few more options.

Once I sent my message I expected at least a dozen different groups within Microsoft to make some kind of offer that might keep me in the company. But I actually received nothing but congratulations and well-wishes. Really! Less than a year earlier I probably could have approached almost any group manager and found an immediate position in his or her team. But now there wasn't a single word in any response that even so much as *hinted* at my staying on. It was so strange…but then again, strange things happen when the Universal Harmony is writing the script!

Speaking of which, consider this one: I needed to decide when to exercise all my stock options since they would automatically expire within ninety days of my November retirement. "Should I wait until the new year to gain the tax benefits?" I asked myself. "Or should I just exercise them all now in case there's a big market crash?" (The stock market was already becoming "irrationally exuberant" at the time.) "And what should my target price be?" Again I prayed incessantly to be guided to do the right thing for myself.

Ooops—I should've said "for everyone"—God apparently felt that "the right thing for me" meant a good lesson in You Get What You Pray For! You see, I was planning to being tithing,

as a spiritual practice, ten percent of all my income to charitable causes in the new year.* To wait until January to cash out meant giving away a pretty substantial chunk of change; more, in fact, than I felt comfortable with. So seeing that there was a little selfishness in my heart, God inspired me, if only by the degree of my own fear, to exercise my options in October when Microsoft stock was already at an all-time high of $136 a share. Seemed prudent! Well, in December, easily within the period when I could have still held my options, the stock split 2-for-1 and immediately shot up to $90 (the equivalent of $180 a share before the split). Had I been "inspired" to wait (had I prayed for the benefit of everyone!) I would've not only been able to make a very meaningful donation but would have also ended up with a whole lot more for myself. Ouch! I think this has got to be the most expensive spiritual lesson I've ever learned.

Another experience of this sort—and one that didn't cost me anything but a little pride, fortunately—came in my very last week with Microsoft. A few months earlier I'd been invited by a summer intern from the University of Pennsylvania to give a talk to his group of computer science students—who called themselves "The Dining Philosophers"—sometime in October. As I was slated to give yet one more talk on OLE in Washington D.C. about the same time, it was a simple matter for me to pass through Philadelphia on the way. My only condition was that I could talk about anything I wanted and that it didn't have to be technical.

---

* According to various surveys, most people say that they would be happy if they only had 10% more money than they do now. That 10% represents perpetual discontent and thus a form of perpetual slavery. Tithing effectively counters this tendency by consciously affirming the ability to live happily with 10% *less*. It's a very powerful practice that ironically has helped many people to overcome financial worries.

"Creativity in the Technical Arts" was probably the most unusual talk I ever gave during my Microsoft career. Drawing on that superconscious experience that led to *Inside OLE 2* (as related in Chapter Ten), I sought to illustrate the truth that creativity is not limited to the traditional forms of painting, music, poetry, and the like: technical disciplines like programming, engineering, and architecture have as much creative potential as any other field. The key is one's ability to attract and receive higher inspiration along with the willingness and energy to manifest that inspiration in some tangible way.

Poor students! I was so enthusiastic about the spiritual teachings I'd so recently discovered that kept finding ways to work them into my talk—rather blatantly at times. In my own joy I just wanted to share so much. In the process, of course, I went a little overboard.

Fortunately I didn't have a large audience with which to embarrass myself. My talk came on the last Friday before the 1996 presidential election and it just so happened that President Clinton himself was leading a rally—on campus—at the exact same time! Secret Service personnel were all over the place, even on the rooftops across the street from my hotel room. Cars and busses jammed the streets. Music and stomping and shouting could be heard a mile away! So instead of having the usual fifty to seventy students for my talk, I only had twenty. And when the "Dining" contingent of the Philosophers took me out that evening, they were down to two: the student who had invited me, and one other. Certainly not the normal reception for guest speakers!

But that one other student was deeply interested in the spiritual aspects of my talk and was asking all kinds of questions before we even got to the restaurant. So of course I told him everything that was happening in my own life. I told him about the particular things I had found, gave him a list of

books he might want to read, and continued talking with him via email in the weeks ahead. Within a few months he resolutely began to draw spiritual principles into his life—even as he went to M.I.T. for graduate school—and has been one of my dearest friends ever since. Just for him I had to make a fool of myself in a talk given opposite the President…God certainly has a sense of humor!

In any case, on November 8th, 1996, at the ripe old age of twenty-eight, I quietly faded out of the Microsoft scene and into the light of a new life. I departed without a fuss, without a tear, and without a single shred of anger or regret in my heart. Indeed, as I turned in my cardkey and left my comfortable little office in Redmond for the last time, I felt only gratitude and love. I had served Microsoft to the best of my abilities, and Microsoft, for its part, had blessed me in ways that I was only able to fully appreciate by writing this book. For Microsoft was the agent through which an ordinary life lived actively for worldly gain was transformed into an equally active life lived for God alone.

Microsoft had certainly given me a great gift.

And no other gift is quite so precious.

EPILOGUE

There's an amusing little illustration in Richard Brodie's book, *Getting Past OK*. On one side, near the bottom, there's the face of a man with his hair standing on end, his eyes bulging wide, and his mouth expressing utter shock. On top of the other side is a big cloud saying to him, "Your purpose in life is to memorize every episode of Gilligan's Island!" The caption for this scene reads, "If a voice from the clouds suddenly told you what to do, would you believe it?"

As outrageous as this example is, it clearly expresses what many people expect when they speak of Divine Guidance, God's Will, and the like. "If God really exists," they say, "would he not make himself obvious? Wouldn't he just appear to us all and tell us what to do?"

Well, if Biblical history is reliable we can say that God tried this back in the time of Moses. He sent plagues upon Egypt to free the Israelites from slavery. He parted the Red Sea, showered manna from Heaven, and appeared as a column of smoke by day and a pillar of fire by night. Then he made a bush burn without really burning it and told everyone exactly what was best for them with the Ten Commandments and a few hundred lesser laws. But given the general failure to even follow the

first ten—which are about as clear and direct as you can get!—it hardly needs mention that humanity as a whole has pretty well botched it.

God, of course, could try all this again. He could appear in the clouds and threaten to flood us out unless we all learned to behave ourselves and love him and love each other like we're supposed to. But then what choice would we really have? It'd be no better than what modern psychology calls codependence: any love and obedience that we'd be able to muster under such conditions would disappear the moment the threats were even slightly relaxed.

No, God wants us to love of our own free will, not because we're given no other alternative. He won't come and say "I am God. You must obey me"; in his humbleness God has vowed himself to silence. If we choose to ignore him and look upon his creation as a meaningless jumble of mindless sub-atomic particles, that's fine. He won't impose himself on us. He'll just let us play in the world, with all its joys and sorrows, as long as we like.

But if we choose to sincerely seek God in some fashion, no matter how insignificant it may seem, he *will* respond. If we seek him as truth and justice, he will guide us accordingly; if we seek him as love and joy, appropriate experiences will come our way. It doesn't matter how we seek God, what matters is that we seek him. The smallest effort on our part will come back a thousand-fold.

In this book I have offered my personal testimony—experience, not belief—as proof of these claims. God responded to my poor efforts in wonderful and sometimes miraculous ways. In each case, he was always *inviting* me to take another step forward, as he has continued to do to this day (but that's another story in itself). Yes, his invitations were usually quite strong, amounting in some cases to a firm nudge in the right direction,

if not a swift kick in the pants! Still, I was always, in every circumstance, free to choose differently. I didn't have to accept my starting job in Developer Support, I didn't have to transfer to Developer Relations, I didn't have to write *Inside OLE 2,* and I didn't have to leave Microsoft when I did…although I think that God would have found other ways to lead me along no matter how often I ignored his prompts.

The difference, though, would have been one of greater pain versus greater delight. The more we try to push God away—to cut ourselves off from the greater reality around us, no matter what we call it—the more we experience pain. The more we can embrace that reality, on the other hand, the more we experience joy no matter what happens to befall us.

In closing, then, I leave you once more with the thought that through whatever form of God draws your devotion—as Heavenly Father, Divine Mother, some more universal aspect like Peace or Joy, or even God as manifest through the life of a great saint or master—consciously make your awareness of the Divine a greater part of your life. Cultivate love in the depths of your heart. Share your thoughts. Listen within for a response. Listen for what's really trying to happen in your life and give yourself wholeheartedly into that flow. For that Spirit is there to lovingly guide us in every moment: in our homes, in our leisure activities, in our relationships, and even in the corporate halls of high technology.

About the Author

Kraig Brockschmidt was born and raised in the Seattle suburb of Renton, Washington. In 1979, when he was 11 years old, his father, an electrical engineer at Boeing, bought him his first computer: a Radio Shack TRS-80 Color Computer. At the same time, his father refused to buy any software. "That," he said, "you will have to write yourself."

Kraig did just that, taking to computer programming with the same passion that his older brother took to art. By 1984, during his sophomore year in high school, Kraig was writing his own software and selling it through various Color Computer magazines. He also published several articles in those same journals, eventually having a regular column in one of them. In this he found that he loved sharing ideas about computer programming as much as the programming itself.

Kraig entered the University of Washington in 1986 to pursue a degree in Computer Engineering. He spent some of his free time volunteering for the Microcomputer Support Lab on campus where he got his first exposure both to IBM-style PCs and to the work of customer support. It was based on this experience that he was offered his first real job: an internship in Microsoft's Developer Support department. (His only other employment was two months for a temp agency through which he did damage returns for United Parcel Service.)

During his time in Product Support, Kraig wrote the Calculator program for Windows that is still shipped with the operating system to this day. Following that success, he was hired to work on some of the other Accessory programs of Windows version 3.0. The following summer he was offered a full-time software development internship in which he continued to work part-time up to his graduation in 1990.

Kraig then returned to Developer Support where he honed his skills in both understanding the intricacies of technology and communicating that understanding to others. In less than a year he became one of the most productive engineers in all of Microsoft's Product Support Division.

After only fourteen months with that group, his unique combination of talents brought him into a much broader role. In late 1991 he took a position in Microsoft's technical evangelism group, Developer Relations, where he remained for the bulk of his career.

In Developer Relations, Kraig used his skills to speed the adoption of Microsoft's newest technologies by other software companies. He offered papers and sample programs that demonstrated exactly how to incorporate those technologies into a wide variety of applications and regularly spoke at industry conferences. In addition, he continued to publish articles in magazines such as Microsoft Systems Journal and Windows Programming Journal.

In 1993 Kraig took his work to another level with the publication of *Inside OLE 2* (Microsoft Press). This book became extremely popular and catapulted him to the status of an industry expert. Being in great demand as a lecturer, he traveled far and wide for several years to help people understand Microsoft's key technologies. He was also in great demand within Microsoft as other development teams regularly approached him for help with their designs. Thus he made im-

portant contributions to many of Microsoft's flagship products including Windows, Office, and Internet Explorer.

Late in his career his life began to take a spiritual turn. While Kraig had been raised a Missouri Synod Lutheran, he had set religion aside shortly after joining Microsoft. Through the ensuing years spirituality was little more than an intellectual curiosity; at different times he didn't think about religion at all while at other times he loathed it. Then in 1995 a deep yearning to know truth began to reorient his priorities; by the end of 1996 his life looked completely different. Kraig had retired from Microsoft (with enough assets from stock options to provide a small but adequate income). He and his wife Kristi (who holds a Master's degree in Electrical Engineering) had moved from their large, almost brand-new suburban home to a humble apartment in an intentional spiritual community in Lynnwood, Washington. More recently they have moved to a similar community in Portland, Oregon, to undertake a new phase of their spiritual lives that includes starting a family.

Since that shift in 1996, they have both been dedicated to seeking God rather than worldly success, and to the ideals of non-attachment, service to others, devotion, simplicity, and self-control, specifically as expressed through the teachings of Paramhansa Yogananda, author of the spiritual classic *Autobiography of a Yogi*. For Kraig, this dedication has expressed itself in a wide range of diverse activities—all part, he says "of an expanding self-identity that is reaching out—literally, it seems—to embrace Infinity." When asked now what he does with his time, he simply answers, "Whatever God puts in front of me." If he's responding to a less spiritually-oriented person he'll simply say, "I'm Self-employed." (That's Self, of course, with a *capital* S!)

These activities have included everything from construction (including wiring, plumbing, and welding), writing, music

(various instruments), conducting, singing (both solo and choral in a number of domestic and international concerts), and real-estate to importing, photography, forest management, office management, volunteering, cooking, graphic design, webmastering, consulting (technical and legal), mechanics, retail sales, ministry, and childhood education. In this latter role he even appeared in a program on National Public Radio. Kraig is also a nationally certified Yoga and meditation instructor and has taught a variety of classes and seminars.

In addition to the technical books he authored at Microsoft, he has more recently published *The Harmonium Handbook* (both in the United States and India) and is working on several additional titles such as *Solving Stress* and *Finding Focus*.

Information on these works and other projects is available on his website, www.kraigbrockschmidt.com.

INDEX

Visit the book's website for photos and other extras:

www.mysticmicrosoft.com

Autographed copies of this book are available
from the author for $20 each (shipping included).
Order from the website or send check/money order to:

Kraig Brockschmidt
7410 SW Oleson Road, PMB 389
Portland, OR 97223-7475

(Note: please check the website for current mailing address.)

You're invited to copy, print, and share this book…

It's free and it's legal

As this book is offered freely, readers are encouraged, though not required, to reciprocate in two ways:

(1) Express gratitude to the author by making a monetary contribution to the author's work and/or writing a positive testimonial about the book with permission to use your words in promotional activities. See www.mysticmicrosoft.com for details or write to the author's address on the copyright page.

(2) "Pay it forward" by sharing the book with others and/or making a gift of money or volunteered time to a worthy cause of your choice. So that these gifts do not go unnoticed, please inform the author of your gifts via www.mysticmicrosoft.com (or regular mail) so the website can show the positive contributions that this work has inspired.

Spirituality/Memoir/Technology

Reader Comments:

"...a fascinating book, it was so good that I read the whole book in a weekend." —MF

"I started reading and before long realized that several tasks I had assigned to myself between now and bedtime were going out the window." —RJ

"A pageturner..." —KNB

Where do you go if you want to grow spiritually?

A church? A monastery?
India? Tibet?

What about the heart of high-tech corporate multinationalism?

If you've ever thought that God only works through formal religious or spiritual channels, think again! *Mystic Microsoft* demonstrates that when there's sincerity of heart and a willingness to offer oneself into whatever Life brings, God can (and will) find a way to guide one's inner growth in any setting.

In this fascinating story, one of Microsoft's most visible technology experts during the company's most expansive growth phase (1988-1996) relates how the very circumstances of his career were the vehicle through which he was inwardly transformed. With insight, wit, and colorful anecdotes about life in the world's leading software company, Kraig Brockschmidt illustrates how one's career, such as his experience writing the Windows Calculator and his bestselling book, *Inside OLE 2*, can be a tremendous opportunity for spiritual growth. As he writes, "In the course of my eight and a half years with [Microsoft] I learned and experienced exactly what you would expect from direct [spiritual] training in a monastery or ashram…"

Mystic Microsoft thus offers hope to those who feel that their careers are at odds with their inner aspirations and those who seek to find a deeper meaning in their worldly responsibilities.

ISBN 978-0-6151-4379-8
51895 >
9 780615 143798

US $18.95