Revenge of the Hackers

Eric Steven Raymond

Thyrsus Enterprises [http://www.tuxedo.org/~esr/]

<esr@thyrsus.com>

This is version 1.9

$Date: 2002/08/02 09:02:15 $

Revision History

Revision 1.9	26 August 2000	esr
DocBook conversion.		
Revision 1.8	31 Aug 1999	esr
This version went into the first printed edition.		

The eruption of open-source software into the mainstream in 1998 was the revenge of the hackers after 20 years of marginalization. I found myself semi-accidentally cast as chief rabble-rouser and propagandist. In this essay, I describe the tumultuous year that followed, focusing on the media stategy and language we used to break through to the Fortune 500. I finish with a look at where the trend curves are going.

Table of Contents

Revenge of the Hackers

I wrote the first version of *A Brief History of Hackerdom* in 1996 as a web resource. I had been fascinated by hacker culture *asas* a culture for many years, since long before I edited the first edition of *The New Hacker's Dictionary* in 1990. By late 1993, many people (including myself) had come to think of me as the hacker culture's tribal historian and resident ethnographer. I was comfortable in that role.

At that time, I had not the faintest idea that my amateur anthropologizing could itself become a significant catalyst for change. I think nobody was more surprised than I when that happened. But the consequences of that surprise are still reverberating through the hacker culture and the technology and business worlds today.

In this essay, I'll recapitulate from my personal point of view the events that immediately led up to the January 1998 "shot heard 'round the world" of the open-source revolution. I'll reflect on the remarkable distance we've come since. Then I will tentatively offer some projections into the future.

Beyond Brooks's Law

My first encounter with Linux came in late 1993, via the pioneering Yggdrasil CD-ROM distribution. By that time, I had already been involved in the hacker culture for fifteen years. My earliest experiences had been with the primitive ARPAnet of the late 1970s; I was even briefly a tourist on the ITS machines. I had already been writing free software and posting it to Usenet before the Free Software Foundation was launched in 1984, and was one of the FSF's first contributors. I had just published the second edition of "The New Hacker's Dictionary". I thought I understood the hacker culture—and its limitations—pretty well.

As I have written elsewhere, encountering Linux came as a shock. Even though I had been active in the hacker culture for many years, I still carried in my head the unexamined assumption that hacker amateurs, gifted though they might be, could not possibly muster the resources or skill necessary to produce a usable multitasking operating system. The HURD developers, after all, had been evidently failing at this for a decade.

But where they failed, Linus Torvalds and his community succeeded. And they did not merely fulfill the minimum requirements of stability and functioning Unix interfaces. No. They blew right past that criterion with exuberance and flair, providing hundreds of megabytes of programs, documents, and other resources. Full suites of Internet tools, desktop-publishing software, graphics support, editors, games...you name it.

Seeing this feast of wonderful code spread in front of me as a working system was a much more powerful experience than merely knowing, intellectually, that all the bits were probably out there. It was as though for years I'd been sorting through piles of disconnected car parts—only to be suddenly confronted with those same parts assembled into a gleaming red Ferrari, door open, keys swinging from the lock and engine gently purring with a promise of power...

The hacker tradition I had been observing for two decades seemed suddenly alive in a vibrant new way. In a sense, I had already been made part of this community, for several of my personal free-software projects had been added to the mix. But I wanted to get in deeper...because every delight I saw also deepened my puzzlement. It was too good!

The lore of software engineering is dominated by Brooks's Law, articulated in Fred Brook's classic *The Mythical Man-Month*. Brooks predicts that as your number of programmers N rises, work performed scales as N but complexity and vulnerability to bugs rises as N^2. N^2 tracks the number of communications paths (and potential code interfaces) between developers' code bases.

Brooks's Law predicts that a project with thousands of contributors ought to be a flaky, unstable mess. Somehow the Linux community had beaten the N^2 effect and produced an OS of astonishingly high quality. I was determined to understand how they did it.

It took me three years of participation and close observation to develop a theory, and another year to test it experimentally. And then I sat down and wrote *The Cathedral and the Bazaar* to explain what I had seen.

Memes and Mythmaking

What I saw around me was a community that had evolved the most effective software-development method ever *and didn't know it!and didn't know it!*. That is, an effective practice had evolved as a set of customs, transmitted by imitation and example, without the theory or language to explain why the practice worked.

In retrospect, lacking that theory and that language hampered us in two ways. First: we couldn't think systematically about how to improve our own methods. Second: we couldn't explain or sell the method to anyone else.

At the time, I was thinking about only the first effect. My sole intention in writing the original paper was to give the hacker culture an appropriate language to use internally, to explain itself to itself. So I wrote down what I had seen, framed as a narrative and with appropriately vivid metaphors to describe the logic that could be deduced behind the customs.

There was no really fundamental discovery in *The Cathedral and the Bazaar*. I did not invent any of the methods it describes. What was novel was not the facts it described but those metaphors and the narrative—a simple, powerful story that encouraged the reader to see the facts in a new way. I was attempting a bit of memetic engineering on the hacker culture's generative myths.

I first gave the full paper at Linux Kongress, May 1997 in Bavaria. The fact that it was received with rapt attention and thunderous applause by an audience in which there were very few native speakers of English seemed to confirm that I was onto something. But, as it turned out, the sheer chance that I was seated next to publisher Tim O'Reilly at the Thursday night banquet set in motion a more important train of consequences.

As a long-time admirer of O'Reilly's institutional style, I had been looking forward to meeting Tim for some years. We had a wide-ranging conversation (much of it exploring our common interest in classic science fiction) that led to an invitation for me to deliver *The Cathedral and the Bazaar* at Tim's Perl Conference later in the year.

Once again, the paper was well-received—with cheers and a standing ovation, in fact. I knew from my email that since Bavaria, word about *The Cathedral and the Bazaar* had spread over the Internet like a fire in dry grass. Many in the audience had already read it, and my speech was less a revelation of novelty for them than an opportunity to celebrate the new language and the consciousness that went with it. That standing ovation was not so much for my work as for the hacker culture itself—and rightly so.

Though I didn't know it, my experiment in memetic engineering was about to light a bigger fire. Some of the people for whom my speech was genuinely novel were from Netscape Communications, Inc. And Netscape was in trouble.

Netscape, a pioneering Internet-technology company and Wall Street highflier, had been targeted for destruction by Microsoft. Microsoft rightly feared that the open Web standards embodied by Netscape's browser might lead to an erosion of the Redmond giant's lucrative monopoly on the PC desktop. All the weight of Microsoft's billions, and shady tactics that would later trigger an antitrust lawsuit, were deployed to crush the Netscape browser.

For Netscape, the issue was less browser-related income (never more than a small fraction of their revenues) than maintaining a safe space for their much more valuable server business. If Microsoft's Internet Explorer achieved market dominance, Microsoft would be able to bend the Web's protocols away from open standards and into proprietary channels that only *Microsoft'sMicrosoft's* servers would be able to service.

Within Netscape there was intense debate about how to counter the threat. One of the options proposed early on was to throw the Netscape browser source open—but it was a hard case to argue without strong reasons to believe that doing so would prevent Internet Explorer dominance.

I didn't know it at the time, but *The Cathedral and the Bazaar* became a major factor in making that case. Through the winter of 1997, as I was working on the material for my next paper, the stage was being set for Netscape to break the rules of the proprietary game and offer my tribe an unprecedented opportunity.

The Road to Mountain View

On 22 January 1998 Netscape announced that it would release the sources of the Netscape client line to the Internet. Shortly after the news reached me the following day, I learned that CEO Jim Barksdale was describing my work to national-media reporters as "fundamental inspiration" for the decision.

This was the event that commentators in the computer trade press would later call "the shot heard 'round the world'—and Barksdale had cast me as its Thomas Paine, whether I wanted the role or not. For the first time in the history of the hacker culture, a Fortune 500 darling of Wall Street had bet its future on the belief that *our way was right.our way was right.* And, more specifically, that *my analysismy analysis* of 'our way' was right.

This is a pretty sobering kind of shock to deal with. I had not been very surprised when *The Cathedral and the Bazaar* altered the hacker culture's image of itself; that was the result I had been trying for, after all. But I was astonished (to say the least) by the news of its success on the outside. So I did some very hard thinking in first few hours after word reached me. About the state of Linux and the hacker community. About Netscape. And about whether I, personally, had what it would take to make the next step.

It was not difficult to conclude that helping Netscape's gamble succeed had just become a very high priority for the hacker culture, and thus for me personally. If Netscape's gamble failed, we hackers would probably find all the opprobrium of that failure piled on our heads. We'd be discredited for another decade. And that would be just too much to take.

By this time I had been in the hacker culture, living through its various phases, for twenty years. Twenty years of repeatedly watching brilliant ideas, promising starts, and superior technologies crushed by slick marketing. Twenty years of watching hackers dream and sweat and build, too often only to watch the likes of the bad old

IBM or the bad new Microsoft walk away with the real-world prizes. Twenty years of living in a ghetto—a fairly comfortable ghetto full of interesting friends, but still one walled in by a vast and intangible barrier of mainsteam prejudice inscribed “ONLY FLAKES LIVE HERE”.

The Netscape announcement cracked that barrier, if only for a moment; the business world had been jolted out of its complacency about what ‘hackers’ are capable of. But lazy mental habits have huge inertia. If Netscape failed, or perhaps even if they succeeded, the experiment might come to be seen as a unique one-off not worth trying to repeat. And then we’d be back in the same ghetto, walls higher than before.

To prevent that, we needed Netscape to succeed. So I considered what I had learned about bazaar-mode development, and called up Netscape, and offered to help with developing their license and in working out the details of the strategy. In early February I flew to Mountain View at their request for seven hours of meetings with various groups at Netscape HQ, and helped them develop the outline of what would become the Mozilla Public License and the Mozilla organization.

While there, I met with several key people in the Silicon Valley and national Linux community. While helping Netscape was clearly a short-term priority, everybody I spoke with had already understood the need for some longer-term strategy to follow up on the Netscape release. It was time to develop one.

The Origins of ‘Open Source’

It was easy to see the outlines of the strategy. We needed to take the pragmatic arguments I had pioneered in *The Cathedral and the Bazaar*, develop them further, and push them hard, in public. Because Netscape itself had an interest in convincing investors that its strategy was not crazy, we could count on it to help the promotion. We also recruited Tim O’Reilly (and through him, O’Reilly & Associates) very early on.

The real conceptual breakthrough, though, was admitting to ourselves that what we needed to mount was in effect a *marketing campaignmarketing campaign*—and that it would require marketing techniques (spin, image-building, and rebranding) to make it work.

Hence the term ‘open source’, which the first participants in what would later become the Open Source campaign (and, eventually, the Open Source Initiative organization) invented at a meeting held in Mountain View the offices of VA Research on 3 February 1998.

It seemed clear to us in retrospect that the term ‘free software’ had done our movement tremendous damage over the years. Part of this stemmed from the fact that the word ‘free’ has two different meanings in the English language, one suggesting a price of zero and one related to the idea of liberty. Richard Stallman, whose Free Software Foundation has long championed the term, says “Think free speech, not free beer” but the ambiguity of the term has nevertheless created serious problems—especially since most “free software” is also distributed free of charge.

Most of the damage, though, came from something worse—the strong association of the term ‘free software’ with hostility to intellectual property rights, communism, and other ideas hardly likely to endear it to an MIS manager.

It was, and still is, beside the point to argue that the Free Software Foundation is not hostile to all intellectual property and that its position is not exactly communistic. We knew that. What we realized, under the pressure of the Netscape release, was that FSF’s actual position didn’t matter. Only the fact that its evangelism had backfired (associating ‘free software’ with these negative stereotypes in the minds of the trade press and the corporate world) actually mattered.

Our success after Netscape would depend on replacing the negative FSF stereotypes with *positivepositive* stereotypes of our own—pragmatic tales, sweet to managers’ and investors’ ears, of higher reliability and lower cost and better features.

In conventional marketing terms, our job was to rebrand the product, and build its reputation into one the corporate world would hasten to buy.

Linus Torvalds endorsed the idea the day after that first meeting. We began acting on it within a few days after. Bruce Perens had the `<opensource.org>` domain registered and the first version of the Open Source website [http://www.opensource.org] up within a week. He also suggested that the Debian Free Software Guidelines become the ‘Open Source Definition [http://www.opensource.org/osd.html]’, and began the process of registering ‘Open Source’ as a certification mark so that we could legally require people to use ‘Open Source’ for products conforming to the OSD.

Even the particular tactics needed to push the strategy seemed pretty clear to me even at this early stage (and were explicitly discussed at the initial meeting). Key themes:

1. Forget Bottom-Up; Work on Top-Down

One of the things that seemed clearest was that the historical Unix strategy of bottom-up evangelism (relying on engineers to persuade their bosses by rational argument) had been a failure. This was naive and easily trumped by Microsoft. Further, the Netscape breakthrough didn’t happen that way. It happened because a strategic decision-maker (Jim Barksdale) got the clue and then imposed that vision on the people below him.

The conclusion was inescapable. Instead of working bottom-up, we should be evangelizing top-down—making a direct effort to capture the CEO/CTO/CIO types.

2. Linux is Our Best Demonstration Case

Promoting Linux must be our main thrust. Yes, there are other things going on in the open-source world, and the campaign will bow respectfully in their direction—but Linux started with the best name recognition, the broadest software base, and the largest developer community. If Linux can’t consolidate the breakthrough, nothing else will, pragmatically speaking, have a prayer.

3. Capture the Fortune 500

There are other market segment that spend more dollars (small business and home office being the most obvious examples) but those markets are diffuse and hard to address. The Fortune 500 doesn’t merely *havehave* lots of money, it concentrates lots of money where it’s relatively accessible. Therefore, the software industry largely does what the Fortune 500 business market tells it to do. And therefore, it is primarily the Fortune 500 we need to convince.

4. Co-opt the Prestige Media that Serve the Fortune 500

The choice to target the Fortune 500 implies that we need to capture the media that shape the climate of opinion among top-level decision-makers and investors: very specifically, the *New York Times*, the *Wall Street Journal*, the *Economist*, *Forbes*, and *Barron's Magazine*.

On this view, co-opting the technical trade press is necessary but not sufficient; it's important essentially as a pre-condition for storming Wall Street itself via the elite mainstream media.

5. Educate Hackers in Guerrilla Marketing Tactics

It was also clear that educating the hacker community itself would be just as important as mainstream outreach. It would be insufficient to have one or a handful of ambassadors speaking effective language if, at the grass roots, most hackers were making arguments that didn't work.

6. Use the Open Source Certification Mark to Keep Things Pure

One of the threats we faced was the possibility that the term 'open source' would be "embraced and extended" by Microsoft or other large vendors, corrupting it and losing our message. It is for this reason the Bruce Perens and I decided early on to register the term as a certification mark and tie it to the Open Source Definition (a copy of the Debian Free Software Guidelines). This would allow us to scare off potential abusers with the threat of legal action.

It eventually developed that the U.S. Patent and Trademark office would not issue a trademark for such a descriptive phrase. Fortunately, by the time we had to write off the effort to formally trademark "Open Source" a year later, the term had acquired its own momentum in the press and elsewhere. The sorts of serious abuse we feared have not (at least, not yet as of November 2000) actually materialized.

The Accidental Revolutionary

Planning this kind of strategy was relatively easy. The hard part (for me, anyway) was accepting what my own role had to be.

One thing I understood from the beginning is that the press almost completely tunes out abstractions. They won't write about ideas without larger-than-life personalities fronting them. Everything has to be story, drama, conflict, sound bites. Otherwise, most reporters will simply go to sleep—and even if they don't, their editors will.

Accordingly, I knew somebody with very particular characteristics would be needed to front the community's response to the Netscape opportunity. We needed a firebrand, a spin doctor, a propagandist, an ambassador, an evangelist—somebody who could dance and sing and shout from the housetops and seduce reporters and huggermug with CEOs and bang the media machine until its contrary gears ground out the message: *the revolution is here!the revolution is here!*.

Unlike most hackers, I have the brain chemistry of an extrovert and had already had extensive experience at dealing with the press. Looking around me, I couldn't see anyone better qualified to play evangelist. But I didn't want the job, because I knew it would cost me my life for many months, maybe for years. My privacy would be destroyed.

I'd probably end up both caricatured as a geek by the mainstream press and (worse) despised as a sell-out or glory-hog by a significant fraction of my own tribe. Worse than all the other bad consequences put together, I probably wouldn't have time to hack any more!

I had to ask myself: are you fed up enough with watching your tribe lose to do *whatever it takeswhatever it takes* to win? I decided the answer was yes—and having so decided, threw myself into the dirty but necessary job of becoming a public figure and media personality.

I'd learned some basic media chops while editing *The New Hacker's Dictionary*. This time I took it much more seriously and developed an entire theory of media manipulation, which I then proceeded to apply. The theory centers around the use of what I call "attractive dissonance" to fan an itchy curiosity about the evangelist, and then exploiting that itch for all it's worth in promoting the ideas.

This is not the place for a detailed exposition of my theory. But intelligent readers can probably deduce much of it from the phrase "optimal level of provocation" and the fact that my interview technique involves cheerfully discussing my interests in guns, anarchism and witchcraft while looking as well-groomed, boyishly charming, and all-American wholesome as I can possibly manage. The trick is to sound challengingly weird but convey a reassuring aura of honesty and simplicity. (Note that to make the trick work, I think you have to genuinely *bebe* like that; faking either quality has a high risk of exposure and I don't recommend it.)

The combination of the "open source" label and deliberate promotion of myself as an evangelist turned out to have both the good and bad consequences that I expected. The ten months after the Netscape announcement featured a steady exponential increase in media coverage of Linux and the open-source world in general. Throughout this period, approximately a third of these articles quoted me directly; most of the other two thirds used me as a background source. At the same time, a vociferous minority of hackers declared me an evil egotist. I managed to preserve a sense of humor about both outcomes (though occasionally with some difficulty).

My plan from the beginning was that, eventually, I would hand off the evangelist role to some successor, either an individual or organization. There would come a time when charisma became less effective than broad-based institutional respectability (and, from my own point of view, the sooner the better!). I am attempting to transfer my personal connections and carefully built-up reputation with the press to the Open Source Initiative, an incorporated nonprofit formed specifically to manage the Open Source trademark. At time of writing I am still the president of this organization, but hope and expect not to remain so indefinitely.

Phases of the Campaign

The open-source campaign began with the Mountain View meeting, and rapidly collected an informal network of allies over the Internet (including key people at Netscape and O'Reilly Associates). Where I write 'we' below I'm referring to that network.

From 3 February to around the time of the actual Netscape release on 31 March, our primary concern was convincing the hacker community that the 'open source' label and the arguments that went with it represented our best shot at persuading the mainstream. As it turned out, the change was rather easier than we expected. We discovered a lot of pent-up demand for a message less doctrinaire than the Free Software Foundation's.

Tim O'Reilly invited twenty-odd leaders of major free software projects to what came to be called the Free Software Summit on 7 March. When these leaders voted to adopt the term 'open source', they formally ratified a trend that was already clear at the grass roots among developers. By six weeks after the Mountain View meeting, a healthy majority of the community was speaking our language.

The publicity following the Free Software Summit introduced the mainstream press to the term, and also gave notice that Netscape was not alone in adopting the open-source concept. We'd given a name to a phenomenon whose impact was already larger than anyone outside the Internet community had yet realized. Far from being fringe challengers, open source programs were already market leaders in providing key elements of the Internet infrastructure. Apache was the leading web server, with more than 50% market share (now grown to more than 60%.) Perl was the dominant programming language for the new breed of web-based applications. Sendmail routes more than 80% of all Internet email messages. And even the ubiquitous domain name system (which lets us use names like www.yahoo.com rather than obscure numeric IP addresses) depends almost entirely on an open-source program called BIND. As Tim O'Reilly said during the press conference following the summit, pointing to the assembled programmers and project leaders: "These people have created products with dominant market share using only the power of their ideas and the networked community of their co-developers." What more might be possible if large companies also adopted the open source methodology?

That was a good start to our 'air war', our attempt to change perceptions through the press. But we still needed to maintain momentum on the ground. In April, after the Summit and the actual Netscape release, our main concern shifted to recruiting as many open-source early adopters as possible. The goal was to make Netscape's move look less singular—and to buy us insurance in case Netscape executed poorly and failed its goals.

This was the most worrying time. On the surface, everything seemed to be coming up roses; Linux was moving technically from strength to strength, the wider open-source phenomenon was enjoying a spectacular explosion in trade press coverage, and we were even beginning to get positive coverage in the mainstream press. Nevertheless, I was uneasily aware that our success was still fragile. After an initial flurry of contributions, community participation in Mozilla was badly slowed down by its requirement for the proprietary Motif toolkit. None of the big independent software vendors had yet committed to Linux ports. Netscape was still looking lonely, and its browser still losing market share to Internet Explorer. Any serious reverse could lead to a nasty backlash in the press and public opinion.

Our first serious post-Netscape breakthrough came on 7 May when Corel Computer announced its Linux-based Netwinder network computer. But that wasn't enough in itself; to sustain the momentum, we needed commitments not from hungry second-stringers but from industry leaders. Thus, it was the mid-July announcements by Oracle and Informix that really closed out this vulnerable phase.

The database outfits joined the Linux party three months earlier than I expected, but none too soon. We had been wondering how long the positive buzz could last without major ISV support and feeling increasingly nervous about where we'd actually find that. After Oracle and Informix announced Linux ports other ISVs began announcing Linux support almost as a matter of routine, and even a failure of Mozilla became survivable.

Mid-July through the beginning of November was a consolidation phase. It was during this time that we started to see fairly steady coverage from the financial media I had originally targeted, led off by articles in *The Economist* and a cover story in *Forbes*. Various hardware and software vendors sent out feelers to the open-source community and began to work out strategies for getting advantage from the new model. And internally, the biggest closed-source vendor of them all was beginning to get seriously worried.

Just *howhow* worried became apparent when the now-infamous Halloween Documents [http://-www.opensource.org/halloween/] leaked out of Microsoft. These internal strategy documents recognized the power of the open source model, and outlined Microsoft's analysis of how to combat it by corrupting the open protocols on which open source depends and choking off customer choice.

The Halloween Documents were dynamite. They were a ringing testimonial to the strengths of open-source development from the company with the most to lose from Linux's success. And they confirmed a lot of peoples' darkest suspicions about the tactics Microsoft would consider in order to stop it.

The Halloween Documents attracted massive press coverage in the first few weeks of November. They created a new surge of interest in the open-source phenomenon, serendipitously confirming all the points we had been making for months. And they led directly to a request for me to confer with a select group of Merrill Lynch's major investors on the state of the software industry and the prospects for open source. Wall Street, finally, came to us.

The following six months were a study in increasingly surreal contrasts. On the one hand, I was getting invited to give talks on open source to Fortune 100 corporate strategists and technology investors; for the first time in my life, I got to fly first class and saw the inside of a stretch limousine. On the other hand, I was doing guerrilla street theater with grass-roots hackers—as in the riotously funny Windows Refund Day demonstration of 15 March 1999, when a band of Bay-area Linux users actually marched on the Microsoft offices in the glare of full media coverage, demanding refunds under the terms of the Microsoft End User License for the unused Windows software that had been bundled with their machines.

I knew I was going to be in town that weekend to speak at a conference hosted by the Reason Foundation, so I volunteered to be a marshal for the event. Back in December I'd been featured in a Star Wars parody plot [http://www.userfriendly.org/cartoons/archives/98dec/19981203.html] in the Internet comic strip "User Friendly". So I joked with the organizers about wearing an Obi-Wan Kenobi costume at the demonstration.

To my surprise, when I arrived I found the organizers had actually made a passable Jedi costume—and that's how I found myself leading a parade that featured cheeky placards and an American flag and a rather large plastic penguin, booming out "May the Source be with you!" to delighted reporters. To my further surprise, I was drafted to make our statement to the press.

I suppose none of us should have really been astonished when the video made CNBC. The demonstration was a tremendous success. Microsoft's PR position, still trying to recover from the exposure of the Halloween Documents, took another body blow. And within weeks, major PC and laptop manufacturers began announcing that they would ship machines with no Windows installed and no "Microsoft tax" in the price. Our bit of guerilla theater, it appeared, had struck home.

The Facts on the Ground

While the Open Source campaign's air war in the media was going on, key technical and market facts on the ground were also changing. I'll briefly review some of them here because they combine interestingly with the trends in press and public perception.

In the eighteen months after the Netscape release, Linux continued to grow rapidly more capable. The development of solid symmetric-multiprocessing support and the effective completion of the 64-bit cleanup laid important groundwork for the future.

The roomful of Linux boxes used to render scenes for the Titanic threw a healthy scare into builders of expensive graphics engines. Then the Beowulf supercomputer-on-the-cheap project showed that Linux's Chinese-army sociology could be successfully applied even to cutting-edge scientific computing.

Nothing dramatic happened to vault Linux's open-source competitors into the limelight. And proprietary Unixes continued to lose market share; in fact, by mid-year only NT and Linux were actually gaining market share in the Fortune 500, and by late fall Linux was gaining faster (and more at the expense of NT than of other Unixes).

Apache continued to increase its lead in the web-server market. (By August 1999 Apache and its derivatives would be running fully 61% of the world's publicly-accessible Web servers.) In November 1998, Netscape's browser reversed its market-share slide and began to make gains against Internet Explorer.

In April 1999 the respected computer-market researchers IDG predicted that Linux would grow twice as fast as all other server operating systems combined through 2003—and faster than Windows NT. In May, Kleiner-Perkins (Silicon Valley's leading venture-capital firm) took a lead position in financing a Linux startup.

About the only negative development was the continuing problems of the Mozilla project. I have analyzed these elsewhere (in *The Magic Cauldron*). They came to a head when Jamie Zawinski, a Mozilla co-founder and the public face of the project, resigned a year and a day after the release of the source code, complaining of mismanagement and lost opportunities.

But it was an indication of the tremendous momentum open source had acquired by this time that Mozilla's troubles did not noticeably slow down the pace of adoption. The trade press, remarkably, drew the right lesson: "Open source," in Jamie's now-famous words, "is [great, but it's] not magic pixie dust."

In the early part of 1999 a trend began among big independent software vendors (ISVs) to port their business applications to Linux, following the lead set earlier by the major database vendors. In late July, the biggest of them all, Computer Associates, announced that it would be supporting Linux over much of its product line. And preliminary results from an August 1999 survey of 2000 IT managers revealed that 49% consider Linux an "important or essential" element of their enterprise computing strategies. Another survey by IDC described what it called "an amazing level of growth" since 1998, when the market research couldn't find statistically significant use of Linux; 13% of the respondents now employ it in business operations.

The year 1999 also saw a wave of wildly successful Linux IPOs by Red Hat Linux, VA Linux Systems, and other Linux companies. While the overblown dot-com-like initial valuations investors originally put on them didn't outlast the big market corrections in March 2000, these firms established an unmistakable for-profit industry around open source that continues to be a focus of investor interest.

Into the Future

I have rehearsed recent history here only partly to get it into the record. More importantly, it sets a background against which we can understand near-term trends and project some things about the future.

First, safe predictions for the next year:

- The open-source developer population will continue to explode, a growth fueled by ever-cheaper PC hardware and fast Internet connections.
- Linux will continue to lead the way, the sheer size of its developer community overpowering the higher average skill of the open-source BSD people and the tiny HURD crew.
- ISV commitments to support the Linux platform will increase dramatically; the database-vendor commitments were a turning point.
- The Open Source campaign will continue to build on its victories and successfully raise awareness at the CEO/CTO/CIO and investor level. MIS directors will feel increasing pressure to go with open-source products not from below but from *aboveabove*.
- Stealth deployments of Samba-over-Linux will replace increasing numbers of NT machines even at shops that have all-Microsoft policies.
- The market share of proprietary Unixes will continue to gradually erode. At least one of the weaker competitors (likely DG-UX or HP-UX) will actually fold. But by the time it happens, analysts will attribute it to Linux's gains rather than Microsoft's.
- Microsoft will not have an enterprise-ready operating system, because Windows 2000 will not ship in a usable form. (At 60 million lines of code and still bloating, its development is out of control.)

I wrote the above predictions in mid-December of 1998. All are still holding good as of November 2000, two years after they were written. Only the last one is arguable; Microsoft managed to ship Windows 2000 by drastically curtailing its feature list; adoption rates have not been what they hoped.

Extrapolating these trends certainly suggests some slightly riskier predictions for the medium term (18 to 32 months out).

- Support operations for commercial customers of open-source operating systems will become big business, both feeding off of and fueling the boom in business use.
  (This has already come true in 1999 with the launch of LinuxCare, and Linux support-service announcements by IBM and HP and others.)
- Open-source operating systems (with Linux leading the way) will capture the ISP and business data-center markets. NT will be unable to resist this change effectively; the combination of low cost, open sources, and true 24/7 reliability will prove unstoppable.
- The proprietary-Unix sector will almost completely collapse. Solaris looks like a safe bet to survive on high-end Sun hardware, but most other players' proprietary Unixes will quickly become legacy systems.
  (In early 2000 SGI's IRIX was dead-ended by official Linux adoption within SGI itself, and in mid-2000 SCO agreed to be acquired by Caldera. It now looks probable that a number of Unix hardware vendors will switch horses to Linux without much fuss, as SGI is already well into the process of doing.)

- Windows 2000 will be either canceled or dead on arrival. Either way it will turn into a horrendous train wreck, the worst strategic disaster in Microsoft's history. However, their marketing spin on this failure will be so deft that it will barely affect their hold on the consumer desktop within the next two years.
  (In mid-2000, a just-published IDG survey suggested that "dead on arrival" looks more likely all the time, with most large corporate respondents simply refusing to deploy the initial release and existing deployments experiencing serious security and stability problems. The fact that Microsoft itself was cracked twice in late October/early November of 2000 hardly helped.)

At first glance, these trends look like a recipe for leaving Linux as the last one standing. But life is not that simple, and Microsoft derives such immense amounts of money and market clout from the desktop market that it can't safely be counted out even after the Windows 2000 train wreck.

But there are also reasons to believe that Microsoft is going to experience serious problems in 2001 that aren't related to either Linux or the Department of Justice. As hardware prices drop, the 59% of Microsoft's revenues that come from selling fixed-price preinstallation licenses to PC OEMs is under pressure. Those fixed license costs represent an ever-increasing slice of OEM's gross margins; at some point, the OEMs are going to have to claw back some of that last margin from Redmond in order to make any profits at all. We know where the critical price point is from observing the appliance and PDA market; it's at about $350 retail. On previous trends, desktop prices will cross $350 going down well before midyear 2001—and when that happens, OEMs will have to defect from the Microsoft camp to survive.

Nor will it help Microsoft to respond in the obvious way by charging a percentage of the system's retail price instead of a fixed per-unit fee. OEMs can easily fiddle that system by unbundling expensive outboard components like the monitor—and even if they didn't, Wall Street would regard such a move as an admmission that Microsoft had lost control of its future revenues. One way or another, Microsoft's revenues look likely to crash hard long before DOJ gets a final ruling.

So at two years out the crystal ball gets a bit cloudy. Which of several futures we get depends on questions like: will the DOJ actually succeed in breaking up Microsoft? Might BeOS or OS/2 or Mac OS/X or some other niche closed-source OS, or some completely new design, find a way to go open and compete effectively with Linux's 30-year-old base design? At least Y2K fizzled...

These are all fairly imponderable. But there is one such question that is worth pondering: Will the Linux community actually deliver a good end-user–friendly GUI interface for the whole system?

In the 1999 first edition of this book, I said the most likely scenario for late 2000/early 2001 has Linux in effective control of servers, data centers, ISPs, and the Internet, while Microsoft maintains its grip on the desktop. By November 2000 this prediction had proved out pretty completely except in large corporate data centers, and there it looks very likely to be fulfilled within months.

Where things go from there depend on whether GNOME, KDE, or some other Linux-based GUI (and the applications built or rebuilt to use it) ever get good enough to challenge Microsoft on its home ground.

If this were primarily a technical problem, the outcome would hardly be in doubt. But it isn't; it's a problem in ergonomic design and interface psychology, and hackers have historically been poor at these things. That is, while hackers can be very good at designing interfaces for other hackers, they tend to be poor at modeling the thought processes of the other 95% of the population well enough to write interfaces that J. Random End-User and his Aunt Tillie will pay to buy.

Applications were 1999's problem; it's now clear we'll swing enough ISVs to get the ones we don't write ourselves. I believe the problem for 2001 and later is whether we can grow enough to meet (and *exceedexceed*!) the interface-design quality standard set by the Macintosh, combining that with the virtues of the traditional Unix way.

As of mid-2000, help may be on the way from the inventors of the Macintosh! Andy Hertzfeld and other members of the original Macintosh design team have formed a open-source company called Eazel with the explicit goal of bringing the Macintosh magic to Linux.

We half-joke about 'world domination', but the only way we will get there is by *servingserving* the world. That means J. Random End-User and his Aunt Tillie; and *thatthat* means learning how to think about what we do in a fundamentally new way, and ruthlessly reducing the user-visible complexity of the default environment to an absolute minimum.

Computers are tools for human beings. Ultimately, therefore, the challenges of designing hardware and software must come back to designing for human beings—*allall* human beings.

This path will be long, and it won't be easy. But I think the hacker community, in alliance with its new friends in the corporate world, will prove up to the task. And, as Obi-Wan Kenobi might say, "the Source will be with us".