Why Open Source Makes Sense

By Linus Torvalds and David Diamond
BM is a company with a history of screwing people over. It made its money by getting a captive audience and making sure nobody else got a foothold. That’s how most computer companies worked, in fact. It’s how some of them still do. Then, when IBM developed the personal computer, it unintentionally opened up its technology for anyone to replicate. That single act did more than anything to spur the PC Revolution, which has in turn spurred the Information Revolution, Internet Revolution, New Economy—whatever it is they’re now calling the massive changes taking place throughout the world.

It’s the best illustration of the limitless benefits to be derived from the open source philosophy. While the PC wasn’t developed using the open source model, it is an example of a technology that was opened for any person or company to clone and improve and sell. In its purest form, the open source model allows anyone to participate in a project’s development or commercial exploitation. Linux is obviously the most successful example. What started out in my messy Helsinki bedroom has grown to become the largest collaborative project in the history of the world. It began as an ideology shared by software developers who believed that computer source code should be shared freely, with the General Public License—the anticyclight—as the movement’s powerful tool. It evolved to become a method for the continuous development of the best technology. And it evolved further to gain widespread market acceptance, as seen in the snowballing adoption of Linux as an operating system for Web servers, and in its unexpectedly generous IPOs.

What was inspired by ideology has proved itself as technology and is working in the marketplace. Now open source is expanding beyond the technical and business domains. At Harvard University Law School, professors Larry Lessig (who is now at Stanford) and Charles Nesson have brought the open source model to law. They started the Open Law Project, which relies on volunteer lawyers and law students posting opinions and research to the project’s Web site to help develop arguments and briefs challenging the United States Copyright Extension Act. The theory is that the strongest arguments will be developed when the largest number of legal minds are working on a project, and as a mountain of information is generated through postings and repostings. The site nicely sums up the tradeoff from the traditional approach: “What we lose in secrecy, we expect to regain in depth of sources and breadth of argument.” (Put in another context: With a million eyes, all software bugs will vanish.)

It’s a wrinkle on how academic research has been conducted for years, but one that makes sense on a number of fronts. Think of how this approach could speed up the development of cures for disease, for example. Or how, with the best minds on the task, international diplomacy could be strengthened. As the world becomes smaller, as the pace of life and business intensifies, and as the technology and information become available, people realize the tightfisted approach is becoming increasingly outmoded.

The theory behind open source is simple. In the case of an operating system, the source code—the programming instructions underlying the system—is free. Anyone can improve it, change it, and exploit it. But those improvements, changes, and exploitations have to be made freely available. Think Zen. The project belongs to no one and to everyone. When a project is opened up, there is rapid and continual improvement. With teams of contributors working in parallel, the results can happen far more speedily and successfully than if the work were being conducted behind closed doors.

With teams of contributors working in parallel, the results can happen far more speedily and successfully than if the work were being conducted behind closed doors.

Reprinted by permission of HarperCollins Publishers, Inc. From Just for Fun: The Story of an Accidental Revolutionary, by Linus Torvalds and David Diamond. Copyright © 2001 by Linus Torvalds and David Diamond. Linus Torvalds, the creator of the Linux operating system, currently works as a programmer on several projects for Transmeta. David Diamond has written regularly for such publications as the New York Times, Business Week, and Wired.
That's what we experienced with Linux. Imagine: Instead of a tiny cloistered development team working in secret, you have a monster on your side. Potentially millions of the brightest minds are contributing to a project, and are supported by a peer-review process that has no, er, peer.

The first time people hear about the open source approach, it sounds ludicrous. That's why it has taken years for the message of its virtues to sink in. Ideology isn't what has sold the open source model. It started gaining attention when it was obvious that open source was the best method of developing and improving the highest quality technology. And now it is winning in the marketplace, an accomplishment that has brought open source its greatest acceptance. Companies were able to be created around numerous value-added services, or to use open source as a way of making a technology popular. When the money rolls in, people get convinced.

One of the least understood pieces of the open source puzzle is how so many good programmers would deign to work for absolutely no money. A word about motivation is in order. In a society where survival is more or less assured, money is not the greatest of motivators. It's been well established that folks do their best work when they are driven by a passion. When they are having fun. This is as true for playwrights and sculptors and entrepreneurs as it is for software engineers. The open source model gives people the opportunity to live their passion. To have fun. And to work with the world's best programmers, not the few who happen to be employed by their company. Open source developers strive to earn the esteem of their peers. That's got to be highly motivating.

It seems that Bill Gates doesn't understand this. Is it possible that he's now embarrassed by an off-putting rhetorical question he asked in 1976? “One thing you do is prevent good software from being written. Who can afford to do professional work for nothing?” he wrote in a letter to open source programmers.

In fact, one way to understand the open source phenomenon is to think about how science was perceived by religion so many centuries ago (if not today, by some creatures). Science was originally viewed as something dangerous, subversive, and antiestablishment—basically how software companies sometimes view open source. And just as science wasn't born out of an effort to undermine the religious establishment, open source wasn't conceived in order to detonate the software establishment. It is there to produce the best technology, and to see where it goes.

Science on its own does not make money. It has been the secondary effects of science that create all the wealth. The same goes for open source. It allows the creation of secondary industries that challenge established businesses, much the way the spin-offs of science challenged the church. You find small companies like VA Linux taking advantage of open source and suddenly being able to compete with traditional corporations. In the words of Sir Isaac Newton, standing on the shoulders of giants.

And yes, as open source gains momentum in the world economy, and as its developers earn recognition, they are becoming increasingly bankable as employees. Companies search credit lists, which are traditionally appended to open source software contributions, to determine who is making multiple contributions. And then they instruct their human resources departments to deliver a wheelbarrow full of money and stock options to potential employees. In a previous paragraph I pronounced that money is not the greatest motivator, and, no, I haven't changed my mind in the ensuing sentences. But I must say that money isn't such a bad thing to have as a reward for hard work. It certainly is handy when it comes to filling up the gas tank in my BMW.

Like science itself, open source's secondary effects are endless. It is creating things that until recently were considered impossible, and opening up unexpected new markets. With Linux, as with other open source projects, companies can make their own versions and their own changes, which really isn't possible any other way. It's exciting to realize that just about everything that's ever been done with Linux was not remotely on the radar when we started. It is even taking off in...
China. Traditionally, software development in Asia has primarily been about translating American or European software. Now folks in that part of the world are using Linux to develop their own software. And I'm really proud of the guy who came up to me at Comdex and wanted to show me the gasoline pump that was running Linux. It was a prototype gasoline pump that was running Linux because he wanted to have Web browsers so gasoline customers could go to CNN.com during the three minutes they're waiting for their tanks to fill. Standing on the shoulders of giants.

It's inspiring that people are using technologies like Linux to just make a better gas pump. That sort of innovation is not likely to have happened within the confines of a company, because if you were a company taking Linux to market you would go for the obvious, which right now is the server market or the high-end desktop market. But open source in general allows companies to make their own decisions about what they want to do. So it's Linux in embedded devices. It's TiVo running Linux and the Transmeta Web Slate running Linux and Telephony using Linux. This is how billions of dollars in wealth is being created from open source.

It's like letting the universe take care of itself. By not controlling the technology, you are not limiting its uses. You make it available and people make local decisions—to use it as a launching pad for their own products and services. And while most of those decisions don't make sense in the larger scale of things, they actually work really well. This is not about trying to spread Linux. It's about making Linux available and then letting it spread itself. And this doesn't apply only to Linux. It applies to any project that's open. Open source makes sense.

People don't quibble with the need for free speech. It is a liberty that people have defended with their lives. Freedom is always something you have to defend with your life. But it's also not an easy choice to make initially. And the same is true of openness. You just have to make the decision to be open. It's a difficult stance to take at first, but it actually creates more stability in the end.

Think of politics. If the logic that's applied against open source were applied to government, then we would always have one-party rule. Obviously, a single-party rule is a great deal simpler than our system of multiple parties, the open political system in which much of the world operates. With one party you don't have to worry about getting agreement with other people. The reasoning would follow that government is too important to waste on the give-and-take of openness. For some reason people see the fallacy of this argument as it applies to politics and government, but not as it applies to business. Ironically, in business it makes people nervous.

The arguments a company uses to keep from opening up technology are convincing. That's not how things are done, management says. And it's scary. People are frightened of change, partly because they don't know how it is going to turn out. By sticking to the status quo, a company can make a better judgment of where it will go, and sometimes that seems more important than being hugely successful. These are companies that will be predictably successful instead of being unpredictably really, really, really successful.

It isn't easy for a corporation to turn an existing product into an open source product. There are a host of thorny issues. For one thing, over the course of months or years that it developed the product, the company built up a great deal of internal knowledge. This in-house intellectual property is the company's bread and butter. The organization is unwilling to relinquish the intellectual property that keeps it alive. But also, the very existence of this internal knowledge creates a barrier for outsiders. It discourages them from participating.

But I have seen companies make the move from closed to open. One story is Wapit, a Finnish company creating service and support infrastructure for various interactive devices. This project involved the company's wall-phone-style Web server. For them, the decision to open source their software makes perfect sense. They want to build up their service, but they have to build up their infrastructure first. That requires writing a
lot of software. It's a necessary evil. So instead of viewing it as a decision to make its intellectual property available to others, they look at it this way: The writing of software takes a lot of engineering time, but it isn't something that creates value from being tightly held at the company.

There were a few things working in Wapit's favor. First, it was not a huge project. Second, the decision to open source was made in the early stages of the company's existence. Management reasoned that it had the resources in-house to develop the product, but it wanted to push to have something more than could be created in-house. It also determined that open sourcing is a good way of furthering Wap as a standard for others to work on.

Early in the game, the company asked for my advice and I told them they needed to fight the urge to have decisions made internally. I suggested that if they were having meetings in which decisions should be made, those meetings should not be closed to outsiders. By keeping the decision-making process a company affair, they would run the risk of alienating outsiders, who would have trouble getting past the company's Old Boy's network. That's one of the major problems of establishing and maintaining an open source project from a corporate environment. It's easy to give lip service to open source, but it can unintentionally degenerate into a two-tier society: Us vs. Them. A lot of decisions get made the easy way—sitting at the cafeteria table discussing the options and developing a consensus without ever opening up the discussion to the outside. People from the outside who have valid opinions are essentially voted down by the fact that the decision was already made in the company cafeteria.

This was one of the problems that plagued Netscape in the months following that company's much-heralded decision, in the spring of 1998, to release the source code (called Mozilla) for its next-generation browser. It took a long time before the project truly lived up to its open source promise. There was a

Probably the most vexing thing in the whole process is giving up control and just accepting the fact that outsiders may actually know better. When folks first hear about the possibility of opening up an existing commercial project, they tend to ask the same questions. One question has to do with how people inside the company would feel about the possibility of having an outsider produce work that is better than their own—and having that so publicly noticeable. I think they should feel great about it, and great that they are getting paid for not even doing most of the work. In that regard, open source—or open anything, for that matter—is unforgiving. It shows who can get the job done, who is better. You can't hide behind managers.

Open source is the best way of leveraging outside talent. But you still need to have somebody inside the company who keeps track of the company's needs. That person may not even be the project's leader. In fact, it could be a benefit to the company if someone on the outside takes it over and is doing it for free. It's fine if someone outside is doing a better job. The trouble is, the outsider might lead the project in a direction that doesn't satisfy the company's requirements. So the company must take care of its own needs. The opening of the project might enable the organization to shrink its local resources, but that doesn't mean it can get rid of them. The project could expand to become far bigger than it would have been at a single company. Outside resources make for a cheaper, more complete, and more balanced system, but there's this flip side: The expanded system no longer takes only the company's needs into account. It actually might consider the needs of customers.

Probably the most vexing thing in the whole process is giving up control and just accepting the fact that outsiders may actually know better. The
other difficulty is finding a strong technical leader in the company. This has to be somebody who is trusted by everyone on two levels—both a technical and a political level. It has to be someone who is able to acknowledge the fact that the project may have been flawed from the start. Instead of trying to hide from such problems, the leader must be able to convince everybody that the best thing to do is to go back and start over, which means breaking stuff. It’s not a message people want to hear. However, coming from someone who commands respect, it’s a message people will accept.

Given the nature of office politics and how corporations typically work, the technical lead person would have to be someone with a fairly strong personality. He or she should be somebody who likes working by email and who avoids taking sides. I wouldn’t use the word “liaison,” because that would imply there are two distinct sides here—the inside camp and the outside camp. And that’s not the way it should be. This technical lead person gets paid by the company to do open source. He or she knows, and everybody else knows, that this person is not paid to agree with his counterparts in the company, but simply to do the project. There’s a danger with having the leader too closely associated with the company. Folks might trust his or her technical ability, but not the non-technical judgment.

Is there a diplomat in the house? It’s like “Find me one honest man.”

This is why I have tried so hard over the years to avoid getting involved with Linux companies. This is becoming increasingly critical now that the money is materializing. With so many dollars floating around, people start questioning your motivations. For me, it’s helpful that I’ve been known as being neutral. You have no idea how important it is to me to maintain that neutrality. It drives me nuts.

Okay. You’re right, I should stop preaching. Open source is not for everyone or every project or every corporation. But the more that people start taking stock of the success of Linux, the more they realize this isn’t the knee-jerk rantings of idealistic, unwashed high-schoolers.

Open up anything, and the possibilities will follow. I’ve been talking about open source for as long as journalists have been asking me about it, which is basically the last five years. It used to be that you had to explain and explain what’s so great about it. And, frankly, it felt like an endless trek. It was like trudging in mud.

People get it now.