

How Open Source Software Can Save the ICT Industry One Trillion Dollars per Year

by Michael Tiemann
President Open Source Initiative
Vice President Open Source Affairs, Red Hat

Open Source software is an important^{1,2} and growing³ class of software. Open Source software is distinguished not by programming language, operating environment, nor application domain, but rather by the license(s) that governs the use, distribution, and, most importantly, the rights to access and modify the software's source code. Together, software source code and open source licensing have dramatically changed many conventional assumptions about software and the software industry itself. And change is needed: In 2004 it was estimated that the Linux kernel, if it were to be developed using proprietary methods and private investment, would have cost more than \$600M to create, and that the roughly 1,000 packages typical in Linux distributions of the time would have cost more than \$1B⁴. In 2005 an analysis of the Debian project estimated that its 230 million lines of source code (230M SLOC) would have required 60,000 person-years of development at a cost of more than \$3B USD⁵. But these estimates speak to the possible cost, not the real cost of developing open source (which was far lower, most likely less than 1/20th the cost of developing proprietary software, for reasons discussed later in this article). Nor does it consider the true value of open source, which has the potential to remedy **as much as \$1T USD per year in dead-loss writeoffs of failed proprietary software implementations**. (The \$1T per year figure is extrapolated from the statistics that world-wide ICT is expected to top \$3.4T USD in 2008⁶ and because 18%-30% of all ICT investments, predominantly based on proprietary software because that is the dominant model of the industry today, are dead-loss write-offs^{7,8}).

Before explaining how open source can remedy the \$1T USD problem, it is instructive to look at a contrary view, which is that in 2008 Open Source Software costs the IT industry \$60B per year in lost licensing fees⁹. That is to say that because of open source software, IT customers spent \$60B less than they would have otherwise. Clearly this is a case of glass-half-full/glass-half-empty: the customers who were able to spend less on ICT were able to create more value for themselves¹⁰. We shall now see that this \$60B/year savings is just a small fraction of the total potential savings that open source solutions can deliver on a global basis.

As of early 2009 there are roughly 70¹¹ licenses that are officially recognized as Open Source Licenses. These licenses govern software that spans the software gamut from operating systems (like Linux) and compilers (like GCC and G++) to web servers and clients (like Apache and Firefox) to leading applications and technologies in bioinformatics, statistical display and analysis, 2-D and 3-D paint, illustration and animation software, databases, content management, customer relationship management, accounting, and geographic information systems. And even these categories encompass but a few of the 10,000+ open source software packages that are freely available for major Linux distributions such as Fedora¹², Ubuntu¹³, and Red Hat Enterprise Linux¹⁴.

The documented failure of the industrial model of software development is almost as old as the software industry itself. IBM's OS 360 project, begun in 1961, was perhaps the most ambitious attempt to industrialize the development of software¹⁵. Fred Brooks, the IBM executive responsible for the project¹⁶, hired the best coders, managers, analysts—whatever it took. And IBM ended up spending more money to produce that operating system than the US Government spent on the Manhattan Project to develop the first atomic weapon, a fact that greatly troubled IBM's CEO at the time, Thomas Watson Jr. The product shipped—barely—in 1965, and though the product was a great commercial success, IBM's CEO really wanted to understand how it was possible that the software part of the project was so much more difficult and costly than the hardware part. He put the question to Brooks, but Brooks himself did not have a ready answer. Ten years later Brooks answered the question in the form of a book titled *The Mythical Man Month*. The title comes from

his insight that “it takes a woman nine months to have a child, no matter how many women are assigned the task.” This was the first clue that the process of industrialization—breaking down tasks into simpler steps, optimizing the steps, and then managing the steps through production—had very significant limits when applied to software. Today, this book is often the first reference given to explain why multi-billion dollar software projects like Microsoft's Vista slipped and slipped and slipped again, running wildly over budget and why, when it finally did ship, it did so without the very features that justified the project in the first place. Thus, more thirty years later, we have the Gartner Group (who famously predicted the success of the Personal Computer in enterprise computing) making the only logical prediction that can be inferred from The Mythical Man Month and Microsoft's own experience¹⁷:

Vista will be the last major release of Windows in its current form. The current, integrated architecture of Microsoft Windows is unsustainable - for enterprises and for Microsoft. Each new version integrates more functionality and features, adding to the scale and complexity of the operating system.

While IBM and others struggled to industrialize software, the mainstream industrial model itself was beginning to show weakness across the board. In the 1940s, Dr. W. Edward Deming properly identified serious problems with the model that focused on optimization in terms of measurable (but arbitrary) inputs and outputs as related to costs and production. Deming had no luck convincing American companies to focus on quality as defined by the customer, nor to recognize the value of people working in teams on a larger and transforming goal, nor to consider whether work gave people joy. However, Deming did find a receptive audience in Japan, and the transformation that resulted from his approach has set the standard for modern industry, lowering costs, accelerating innovation, reducing environmental impact, improving safety, and inspiring pride in workmanship¹⁸.

By the 1980s, Deming's work had reached even those still trying to industrialize software, in the context of the Software Engineering Institute (SEI) and their Capability Maturity Model (CMM) in a 1987 SEI Technical Report¹⁹. First adopted not in the US, but offshore, in Chennai, India²⁰, it was credited with dropping software defect density 98% while doubling profits. Though these first attempts have now been retired²¹, the urgent question remains: why is software quality so low²²?

In 2005, Bill Gates gave a keynote speech at the RSA Security conference where he said “We spend over US\$6 billion a year on research and development. I'd say that over a third of that is directly security-focused, and the other two-thirds all tie in and relate to that security work, all the new code being reviewed and going through the threat model, a pretty dramatic thing there”²³. Why is it that companies like Microsoft spend fully a third of their R&D budget correcting remedial security problems that should never have existed in the first place? As President of the Open Source Initiative and Vice President of Open Source Affairs for Red Hat, I believe software quality is low because companies place a higher value on control than on innovation or quality or transparency or user participation. It is a failure to understand both the negative lessons that Brooks have taught, as well as the positive lessons that Deming taught. And perhaps it is because those who have achieved great status, wealth, and power are rarely willing to trade all three for a new system that requires accepting new, disruptive assumptions.

The most significant transformation in ICT has been the emergence of Free and Open Source Software (FOSS).

Consider the popular internet as embodied by the World Wide Web. This vision was first published by Vannevar Bush in 1945²⁴, and prior to 1990 there had been dozens of attempts to create such systems, the most popular being the French MiniTel system²⁵. But none of these systems became truly ubiquitous until the underlying software was free as in freedom. The freedom to read, modify, and share web server and client code led to an explosion of innovation, and to date the most popular product of that explosion, the Apache web server, remains dominant in spite of considerable efforts by some powerful companies to take over that space²⁶.

This came as no surprise to Tim Berners-Lee, creator of the World Wide Web, who explained his decision to make the original web software free and open²⁷:

[H]ad the technology been proprietary, and in my total control, it would probably not have taken off. **The decision to make the Web an open system was necessary for it to be universal.** You can't propose that something be a universal space and at the same time keep control of it.

The popular internet has created a dramatic new factor in the the software industry equation, which is that **developers and users represent a creative continuum**, not distinct castes. Developers can be any people interested in a problem, not merely people employed to work on a specific problem. Free and Open Source software turns on its head the practice of optimizing software production by limiting the number of people working on a problem. Instead, those employed (producers), those annoyed (customers) and those who merely enjoy working on a problem (hackers²⁸) can work together in a problem-specific, rather than organization-specific or property-specific way. In his book *Democratizing Innovation*, Eric Von Hippel presents several studies which, together, show that 85% of all quantum innovation (as opposed to incremental innovation) is user-driven²⁹. **The proprietary software model therefore limits innovation to 1/6th of its theoretical potential.**

The democratization of innovation has also demonstrated a remarkable solution to the problem of *The Mythical Man Month*, thereby transcending the limits of conventional industrialization. For example, *sourceforge.net* is an open source development resource that hosts over 180,000 projects and has more than 1.9M registered users as of December 2008³⁰. Extrapolating from the extensive FLOSS (Free, Libre, and Open Source Software) survey of 2002³¹ (and updated in 2005³²) there were over 490,000 *sourceforge.net* developers in 2006 [when the thesis of this section was first developed--tiemann] who spend more than 10 hours a week or more tending their open source projects³³—an aggregate effort of some 5 million person-hours per week. The three top reasons they list for their involvement is:

1. Because it's fun
2. Because it improves their skills
3. Because it is good for society

Note that this does not include Linux developers (who use *kernel.org*, not *sourceforge.net*), nor Apache, nor the GNU project, nor many of the other larger and more heavily commercialized open source projects. To put these 5 million joy-filled person-hours per week into perspective (again, this does not include Linux, Apache, GNU, or many of the other "large" projects), let's look at the productivity potential of the most successful proprietary software company, Microsoft, in two ways (using numbers that were contemporaneous with the FLOSS survey data, October 2006):

1. If all 61,000 employees³⁴ wrote code, they would have to work over 80 hours/week
2. If Microsoft's \$6.6B/year R&D budget³⁵ were spent on programmers averaging just \$25/hour, they could pay for about 5 million person-hours of work per week

Thus, the *sourceforge.net* website has equaled or exceeded Microsoft's productive potential using a social, not an industrial model. When we consider all the open source developers not included in the *sourceforge.net* numbers (numbers that are increasing exponentially), we see the clear emergence of a new software production capacity entirely outside the conventional limits of the industrial model. Moreover, we find precisely the kind of improvements that Deming would have predicted by taking a transformative approach: according to findings published by Coverity [references below], typical proprietary software has a defect density of 20-30 defects per 1,000 lines of code (KLOC), a number relatively unchanged since the 1960s. When they measured the quality of the Linux kernel (and later, other open source software) they found the following results:

2004: 985 defects in 5.7 MLOC of Linux kernel source code, or **99.3% lower defect density** than average (compare to 114,000 to 171,000 defects in same amount of code)³⁶

2005: While the Linux kernel grew 4.7% in overall code size, defect density decreased by 2.2%. Moreover, **100% of all "serious" defects identified were fixed within 6 months**³⁷

2006: The survey was expanded to entire LAMP stack and an additional 32 OSS programs. No correlation found between size and defect density, implying **OSS development methodology is not limited by scale (nor restricted to just Linux developers)**³⁸

What the top industrialists could not achieve with proprietary software and financial capital, free software has demonstrated with community development and intellectual capital.

But free and open source has done more than revolutionize the production or the quality of software. The open source model has opened innovation. Open source helped crack grand challenge science problems such as sequencing the human genome³⁹. The world's largest public search engine, google.com, is built on open source^{40,41,42}. Open source has become a fixture in tech-heavy disciplines such as financial services⁴³, military intelligence⁴⁴, online retailing⁴⁵, and next-generation cellular telephones and base stations⁴⁶ and handsets, including the recently announced Google Android project/product⁴⁷. And open source is helping to bridge the digital divide for millions of children who are literally off the grid⁴⁸. The model has even informed the strategy of the Bill & Melinda Gates Foundation to better combat infectious diseases⁴⁹:

[On July 20, 2006] the Bill & Melinda Gates Foundation announced that it would require that any researcher who accepts its grant monies for HIV/AIDS research will have to agree to share their scientific findings. The Gates Foundation was apparently frustrated that two decades of secrecy and competition among AIDS researchers have impeded efforts to come up with an AIDS vaccine.

We now returning to the statistic of the \$60B USD that proprietary vendors did not put into their own pockets. In 2006 Open Source software and services earned \$1.8B USD⁵⁰ as compared with \$235B USD in packaged software sales⁵¹ (which likely pulled through an additional \$235B USD in support and services⁵²). No matter how one looks at it, Open Source solutions represent less than 1% of global software spend, and yet now enable a reduction of more than 25% of such spending (because \$60B is more than 25% of \$235B). More impressively, Open Source solutions represent less than 0.1% of global ICT spend, and have already been estimated to deliver back 2% in total returns (\$60B is approx 2% of \$3T). With these kinds of numbers, the idea of spending half one's budget on open source software and half on proprietary becomes meaningless: the whole problem could be solved twice over with Open Source solutions for 10% of what is being spent right now.

At this time, in this moment, the choice is clear: it is time for regime change in the software industry. Open Source software is the key to unlocking 21st century economics and solving 21st century problems. The proprietary software model is unsustainable. It did not scale to meet the challenges of the 1960s, and 50 years later it surely does not scale to address the challenges we now confront. We can no longer afford to pay so much to so few for so little in return. Open Source allows us to invest equally well financial and intellectual capital, and it gives us all the freedom to direct that investment as we each see fit—a scalable model that grows stronger with each additional contributor. The cost savings of Open Source software alone is enough to restore the health of the global economy. When we consider the additional innovation potential that Open Source provides compared to proprietary software, the economic benefits are exponential. Perhaps this explains why an increasing number of local, state, and national governments are adopting formal open source education, development, and procurement policies⁵³. And perhaps we should now have hope that these policies, after years of study, are ready for funding and implementation.

- 1 Hillenius, G., June 26, 2008, EC considers study on migration to Open Source, <http://www.osor.eu/news/eu-ec-considers-study-on-migration-to-open-source/?searchterm=policy>
- 2 Buxbaum, P., Navy to focus only on open systems, March 06, 2008, <http://www.fcw.com/online/news/151858-1.html>
- 3 Deshpande, A. and Riehle, D., The Total Growth of Open Source, March 14, 2008 <http://www.riehle.org/publications/2008/the-total-growth-of-open-source/>
- 4 Wheeler, D., Counting Source Lines of Code (SLOC), June 2001, <http://www.dwheeler.com/sloc/>
- 5 Amor-Iglesias, J., González-Barahona, J., Robles-Martínez, G., Herráiz-Tabernero, I., Measuring Libre Software Using Debian 3.1 (Sarge) as A Case Study: Preliminary Results, <http://www.upgrade-cepis.org/issues/2005/3/up6-3Amor.pdf>
- 6 Mah, P., Gartner: Worldwide IT spending will top \$3 trillion, October 10, 2007, <http://blogs.techrepublic.com.com/tech-news/?p=1348>
- 7 Atwood, J., The Long, Dismal History of Software Project Failure, May 16, 2006, <http://www.codinghorror.com/blog/archives/000588.html>
- 8 Tiemann, M., Software Industry vs. Software Society: Who Wins in 2020, September 2006, <http://people.redhat.com/tiemann/STS-Forum-Tiemann-2006.pdf>
- 9 Rosenberg, D., Study Finds "Free Open Source Software Is Costing Vendors \$60 Billion", April 16, 2008, http://news.cnet.com/8301-13846_3-9920202-62.html
- 10 Tiemann, M., \$60B less for proprietary software = \$60B more customer value, August 23, 2008, <http://www.opensource.org/node/364>
- 11 Open Source Initiative, Licenses By Name, <http://opensource.org/licenses/alphabetical>
- 12 Fedora Project Home Page, <http://fedoraproject.org/>
- 13 Ubuntu Project Home Page, <http://www.ubuntu.com/>
- 14 Red Hat Enterprise Linux Product Page, <http://www.redhat.com/rhel/>
- 15 Wikipedia reference, http://en.wikipedia.org/wiki/The_Mythical_Man-Month
- 16 Frederick Phillips Brooks Biography, Updated June 21, 2006, http://www.cs.unc.edu/~brooks/FPB_BIO.CV.06.2006.pdf
- 17 Moulds, J., Microsoft's Vista release may be last 'big bang', February 1, 2007, <http://www.telegraph.co.uk/finance/2803555/Microsofts-Vista-release-may-be-last-big-bang.html>
- 18 Magnier, M., Rebuilding Japan, October 25, 1999, <http://deming.org/index.cfm?content=651>
- 19 Humphrey, W., Sweet, W., A Method for Assessing the Software Engineering Capability of Contractors, <http://www.sei.cmu.edu/publications/documents/87.reports/87.tr.023.html>
- 20 Business Week Cover Story, Will Bugs Eat Up the U.S. Lead in Software? December 6, 1999, http://www.businessweek.com/1999/99_49/b3658020.htm
- 21 <http://www.sei.cmu.edu/cmml/adoption/migration.html> (retired link, sorry)
- 22 Levinson, M., Let's Stop Wasting \$78 Billion a Year, October 15, 2001, <http://www.cio.com/archive/101501/wasting.html>
- 23 Remarks by Bill Gates, February 15, 2005, <http://www.microsoft.com/presspass/exec/billg/speeches/2005/02-15RSA05.aspx>
- 24 Bush, V., As We May Think, July 1945 <http://www.theatlantic.com/doc/194507/bush>
- 25 Wikipedia reference, <http://en.wikipedia.org/wiki/Minitel>
- 26 Webcraft Web Server Survey, as of January 2009, http://news.netcraft.com/archives/web_server_survey.html
- 27 Tim Berners-Lee Home Page, <http://www.w3.org/People/Berners-Lee/FAQ.html#What2>
- 28 Raymond, E., The Hackers Dictionary, http://www.ccil.org/jargon/jargon_23.html#TAG833
- 29 Von Hippel, E., Democratizing Innovation, 2005, <http://web.mit.edu/evhippel/www/democ1.htm>
- 30 <http://alexandria.wiki.sourceforge.net/What+is+SourceForge.net%3F>
- 31 FLOSS FINAL REPORT, June 2002, http://flossproject.org/floss1/stats_5.html
- 32 FLOSSPOLLS Research Home Page, <http://www.flosspols.org/research.php>
- 33 FLOSS FINAL REPORT, June 2002, http://flossproject.org/floss1/stats_7_2.html
- 34 <http://finance.yahoo.com/q/pr?s=MSFT>
- 35 <http://finance.yahoo.com/q/is?s=MSFT&annual>
- 36 Fisher, D., Linux Kernel Review Shows Far Fewer Flaws, December 14, 2002,

- <http://www.eweek.com/c/a/Linux-and-Open-Source/Linux-Kernel-Review-Shows-Far-Fewer-Flaws/>
- 37 Kerner, S., Study: Linux Code Grows as Defects Decline, August 3, 2005,
<http://www.internetnews.com/dev-news/article.php/3524911>
- 38 Kerner, S., Coverity Study Ranks LAMP Code Quality, March 6, 2006,
<http://www.internetnews.com/stats/article.php/3589361>
- 39 Stein, L., How Perl Saved the Human Genome Project, April 1, 1997,
<http://www.ddj.com/184410424>
- 40 Vaughn-Nichols, S., Google Opens Up About Open Source, October 26, 2005,
<http://www.eweek.com/c/a/Linux-and-Open-Source/Google-Opens-Up-About-Open-Source/>
- 41 http://www.informationweek.com/news/software/open_source/showArticle.jhtml?articleID=187202790
- 42 <http://www.informationweek.com/news/software/linux/showArticle.jhtml?articleID=192300293>
- 43 Galli, P., Open Source Making Inroads on Wall Street, April 24, 2006,
<http://www.eweek.com/c/a/Linux-and-Open-Source/Open-Source-Making-Inroads-on-Wall-Street/>
- 44 <http://www.nsa.gov/selinux/>
- 45 O'Reilly, T., Ask Tim, February 2004,
http://oreilly.com/pub/a/oreilly/ask_tim/2004/amazon_0204.html
- 46 LinuxWorld awash with Linux phone buzz, August 16, 2006,
<http://www.linuxdevices.com/news/NS2577652026.html>
- 47 <http://code.google.com/android/>
- 48 <http://laptop.org/en/>
- 49 Bollier, D., Is Hell Freezing Over? Bill Gates Embraces the Knowledge Commons, July 21, 2006,
<http://onthecommons.org/content.php?id=885>
- 50 Lawton, M., Notarfonzo, R. Worldwide Open Source Software Business Models 2007–2011 Forecast: A Preliminary View. IDC Inc.
- 51 Software & Information Industry Association. Packaged Software Industry Revenue and Growth, 2006. Available from <http://siii.net/software/>
- 52 Author's estimate based on Standish Group and other studies
- 53 Lewis, J., Government Open Source Policies, August 2007, Center for Strategic and International Studies
http://www.csis.org/media/csis/pubs/070820_open_source_policies.pdf