

# Visionen



## Open Source

Ab Seite 4

TechTeam: Paranoia Seite 30

Arbeiten im VIS Seite 40

Ausgabe Oktober 2006



# The 4th Element: Open Source and Globalization

CHRISTOF WITTIG - MIT VISIONEN

Open source software has many facets. One of them is the ability to manage large software projects of globally distributed, individual contributors with virtually no management overhead. The author, a researcher at Stanford's Graduate School of Business and CEO of the open source software company db4objects, argues that this "fourth element of open source" will have an even higher impact on engineering jobs in the Western world than I.T. offshoring, which was so hotly debated during the 2004 U.S. presidential campaign. Though this new form of globalization is still not very widespread, it has the largest economic benefits of several open source practices, which slowly but steadily get adopted by traditional software companies for their production systems. The article concludes with practical advice for forward-thinking software engineers on how to prepare for a changing labor market characterized by global competition and global opportunities.

## The Myth of Non-Commercial Open Source Software

The first wave of open source projects were nearly all driven by somewhat idealistic protagonists such as Richard Stallman, author of the GNU General Public License (GPL), who stipulated that software should be free "to avoid moral corruption."<sup>1</sup> The notion of the long-haired, non-commercial software hacker who works in his free time on cool stuff has found a firm place in the minds of many, although this notion is far from reality.

Today, anywhere from 50 to 100% of the contributors of the most visible open source projects are full-time, dedicated resources on the payroll of open source companies such as Red Hat or OSDL, or of open source backers such as IBM and Oracle. What Intel's Les Valdasz calls "educated conjecture"<sup>2</sup> is the phenomenon that nearly all major open source software projects have started as somewhat non-commercial and, over time, have evolved to projects whose contributions are driven in large part, if not entirely, by paid, dedicated engineers.

What is more important for the industry at large, though, is the fact that closed source software companies have started to heavily adopt the major

elements of open source for their commercial software production processes as well. I distinguish four elements which have made the open source movement so powerful as it is today, establishing products with huge market share such as Apache's 63% vs. Microsoft's 30%<sup>3</sup> :

1. availability of source code
2. free software licenses
3. user-driven, collaborative production
4. global spread of contributing individuals

There is no better evidence for the growing adoption of these open source practices by commercial companies than Microsoft. While Microsoft is traditionally less friendly to open source, it has made significant changes to embrace these practices for commercial purposes: With its SharedSource initiative, Microsoft has started to open the "sacred" source code for Windows and Office to a selected group of users to gather feedback and benefit from their QA. In addition, Microsoft finds itself giving away certain products for free, in order to match open source free or low pricing. Over the past two or three years, you also find plenty of references to "community" on Microsoft's websites and see free peer support emerging, which was introduced by open source projects. All these changes and many more discussions behind the scenes make Redmond-based Ted Neward<sup>4</sup>, speaking at the db4o User Conference in July 2006, speculate that Microsoft will be the largest open source company by 2010.

Besides Microsoft, virtually every forward-thinking software company has added open source elements to key parts in their business model: IBM is leveraging open source software as a competitive strategy<sup>5</sup>. Oracle's Larry Ellison has vowed to "embrace" open source<sup>6</sup>. HP, like many others,

has established a Linux Office. Sun is announcing plans to open source Java (yet doesn't know how). SAP has invested in open source companies MySQL and Zend, and has open sourced its SAP DB (now Max DB). Hardly any major vendor is missing from the list of Eclipse sponsors, including IBM, BEA, HP, Sybase, Wind River, Adobe, Borland, salesforce.com, and so on.

I think it is safe to claim, that today, in 2006, not only all major open source projects are principally commercially driven, but also that most commercial companies have started to incorporate open source elements into their production model.

The myth of non-commercial open source software no longer holds true.

### **The Fourth Element: Globally Distributed Software Development**

At this point in time, conventional software companies still lack most in the adoption of the fourth element of open source: the globally distributed workforce of individual software engineers who work from their homes to contribute to an entirely Internet-enabled production system.

I just spoke to an Engineer who works for IBM on the open source Java "Harmony" project and who is still required to show up in his office in Shanghai every day, though virtually every part of the project is as open source as it gets. The main reasons for this reluctance of traditional software companies to work with distributed teams of individuals are the decrease in managerial influence, the incompatibility with current incentive systems, the dependency on face-to-face communication, and – for closed source companies – the fear to lose control over intellectual property.

Nevertheless, my prediction is that most software product companies will also, sooner or later, adopt the fourth factor. The reason for my belief is economical and simple: This model constitutes a 10x cost advantage over the traditional model of collocated software factories, stemming from:

- the ability to hire the right talent for the right job at much lower cost - anywhere on the globe rather than locally
- the productivity increase from flat and asynchronous Internet-based communication rather than costly hierarchical and synchronous face-to-face interaction, characterized by endless and ineffective meetings
- the empowerment of every brain cell inside and outside of the organization to contribute to a given problem, be it an idea, code, a bug report, or a work-around
- the enablement and adoption of more agile and iterative software development models rather than the waterfall design-write-test approach, which eliminates a lot of misallocations known from so many traditional software projects

Many engineers, who have worked in traditional companies, tell me they find it very difficult to imagine that a distributed model would work: "Being distributed and working from home sounds great, but it will never work in our organization." I agree that most closed-source companies will find it very difficult to change their production and - especially - their communication processes accordingly.

However, backed by my research and my practical experience in running an open source company, I know that it does work, and that these huge cost benefits materialize. According to Darwin, it is therefore just a matter of whether closed source

companies will be able to adopt these practices themselves - or whether they will lose in competition with those who have already successfully adopted them.

### **Case Study: db4objects, Open Source Object Database Company**

This opens the stage to have a closer look at an open source software company as a case study for the future of software production: db4objects Inc. ([www.db4o.com](http://www.db4o.com)) is the company behind db4o, the world's leading open source object database, native to Java and .NET. Started in 2004 out of research at Stanford's Graduate School of Business, it has incorporated the specific properties of existing open source software projects and companies into a coherent business plan right from the outset.

Like MySQL AB<sup>7</sup>, the open source relational database company, db4objects runs a so-called dual license model: From the beginning, any code contribution to the core product distribution has been licensed by the company. As a result, the company is able to distribute the software alternatively under the GPL (for free) or under an affordable commercial license to those who don't want to comply with the constraints of the GPL (such as open sourcing the derivative work). Consequently, nearly all core contributors receive remuneration for their code contributions.

The company is headquartered in San Mateo, CA, in the heart of Silicon Valley. At headquarters, there are only two engineers, but nearly all of the marketing, sales and finance functions. In addition, some 20 contracted software engineers are located around the globe: Based on all 5 continents and in countries as far apart as the U.S., Brazil, South Africa, Germany, U.A.E., China, India, and Australia. Most of the software engineers were hired

from the db4o user community, which results in wide-spread user-advocacy throughout the engineering team. All of them work from home.

The product development is very user-driven. The company hosts the most active forum on object database technology (<http://developer.db4o.com>) and is the launching sponsor of the most up-to-date educational resource on ODBMSs ([www.odbms.org](http://www.odbms.org)). db4o's product roadmap is public and user induced, built on user surveys, user conferences and Wikis. Every two hours, a continuous integration build shares the latest software code with the entire user community of currently 15,000 registered users, which includes customers like Boeing, Bosch Sigpack, Intel and Seagate as well as numerous start-ups, top-tier educational institutions like the MIT, the ETH, and Tokyo University, as well as many non-profit projects.

The development model is agile and based on eXtreme Programming (XP): Pair programming is used to spread knowledge and to produce less error-prone software code. Thanks to the Internet, pair programming in a distributed environment is not a contradiction any more: Free Skype and TightVNC enable pairing over the same IDE between Sao Paolo and Hong Kong, between Mulhouse and Whiteriver - at no cost. Also, there is no head of engineering at db4objects: Tasks are registered by the engineers themselves in the project management system Jira, which is publically accessible. Every peer and every user can track the progress in different parts of the project at any point in time.

### **The Role of the Engineer in an Open Source Company**

There is good news and bad news for software engineers in this model, and certainly a lot of differences to traditional companies.

### **About the author**



Christof Wittig is CEO of the open source software company db4objects ([www.db4o.com](http://www.db4o.com)) and researcher at Stanford's Graduate School of Business.

Christof brings 15 years of experience in software business, sales and marketing to his role as Chief Executive Officer of db4objects. He leads all aspects of the company's business strategy and execution as well as its marketing initiatives.

Christof is leading author of the Stanford's research on MySQL's business model<sup>10</sup>. His interests are to investigate the economic and strategic impacts of open source practices and to find ways how businesses can capitalize on these practices, while strengthening the open source ecosystem.

Christof also serves as President of ODBMS.ORG e.V. ([www.odbms.org](http://www.odbms.org)), a non-profit association to provide free resources on object database technology for education and research. He is a frequent speaker on industry events on open source business and his thoughts have been featured in many publications.

Christof holds a Master of Science in Management from the Stanford Graduate School of Business. He also holds a Master of Engineering from Technical University Munich.

[1] <mailto:christof.wittig@stanford.edu>

The good news first: Open source puts users and software engineers into a much more powerful position than they are with traditional companies, which employ (powerful) gatekeepers in marketing (product marketing) and sales (account management), who tell software developers what to do. Open source companies, on the other side, are based on direct interaction between users and its software engineers, often blending the two in terms of hires or mixed contributions. This eliminates the power brokers in sales and marketing and puts engineers into the driver's seat.

At db4objects, for instance, we spend 70% of our budget on R&D, while most conventional software companies spend only 20% on R&D -- 80% land in the coffers of sales, marketing, professional services, non-productive functions, and managers.

The bad news for (some) software engineers is that they suddenly find themselves in global competition. Especially in high price countries such as the U.S. and Switzerland, software engineers have to think strongly about why (or if) they should get a 5-10x pay differential over a contributor from China, for instance. Local market rates, driven by living standards, lose their significance if a company can select its employees from a fast growing, global pool of talent.

There's another, very big change, which heavily affects the software engineer's life: People work from home, not from offices. They work when they want, not from 9 to 5. They manage themselves and don't report to a boss. What counts is their output: software code which is visible to 1,000s of eyeballs every two hours after checking in some code. It is not the clap on the shoulder by the boss, but the cheers from the user community which count.

### **Globalization 3.0: Effects on the Software Engineer**

Tom Friedman has found very good language to describe the "fourth element" in his book "The World is Flat"<sup>8</sup>. He argues that cheap communication has obliterated barriers to many markets, making more and more of them entirely global and accessible to any individual at very low cost. He consequently speaks about the occurrence of Globalization 3.0, to differentiate the individual going global from the globalization of nation-states since 1492 (1.0) and corporations since the 18th century (2.0). Today, an engineer in a remote hut in Obersaxen in Graubünden, one in Mountain View, California, and one in Chengdu in West China all compete on a level playing field for jobs in companies like db4objects.

Open source's fourth element is a result of these dramatic changes in the ability to communicate and collaborate across the entire globe at virtually no cost. Open source has shown that a globally distributed team of individuals can work together more effectively than co-located engineers, be they in Silicon Valley, Cambridge, or Bangalore. Given the model's economics, the impact on engineering jobs in the U.S. and the Western world will be much more profound than those of offshoring, so heatedly discussed in the 2004 U.S. presidential campaign.

A common response to globalization in the Western world is to reject it, because it means increased competition and price pressure. In my opinion, a better response is to try to understand the phenomenon, to look for its benefits, and to find out how one's self can best add value to the system.

Of course, there are two sides of this coin. On my travels around the globe I meet two distinct groups:

1. Software engineers in below-average-wage countries (such as China, Brazil, and Russia) and
2. Software engineers in above-average-wage countries (such as the U.S., Japan, and Switzerland)

The message of globalization is well received in the first group. For them it is an opportunity to participate with high impact in an exciting industry, to earn far above local wages and thus help entire families and regions to be lifted out of poverty into prosperity. Generations of young and eager software workers in these below-average-wage countries now enter the economy under totally different premises than their parents, who had very little career prospects or access to the appropriate education. Open source provides them with job opportunities - and gives them also access to free education and learning.

While I write this, I am visiting Chengdu, the population 11-million capital of China's Sichuan province, and I have met with talented and eager software engineers who contribute to db4o. The local universities produce some 1,000 computer science graduates a year, and most of them still have difficulty finding a decent job. The average remuneration for a senior engineer is \$700 per month.

Clearly, globalization and open source opportunities are a huge benefit to this first group.

I want to use the remainder of this article to share my thoughts on how the second group, software engineers in high-wage Western countries, can prepare for and benefit from this trend, too.

### **What to Do as a Software Engineer in a High-Wage Country like Switzerland**

In the Western world, the number one fear of globalization is that it equalizes all wages globally, which means a reduction in the standard of living.

However, this would only be true if people were what economists call a "commodity," a product where all units are equal. As we all know, people are different by education, culture, skills, habits, language, location, and more. Some people may like to work on a hard core engineering problem for 3 weeks; others rather prefer to create a slick user interface. Some find that they can write well or speak the same language as key customers; others have a tool or domain expertise which makes them unique.

The first step to position yourself in an intensifying market is therefore to understand what you are good at and where you can add the best value to your industry segment. This way you learn to differentiate yourself from an anonymous pool of software workers.

Open source then acts as your career builder. You can use open source communities to adapt your job profile to fit changing market demands for software engineers. The best way to get your feet wet is to participate in the community of a project which comes closest to your interests. You can start to build related pet projects and share them with your peers. You will learn from their feedback, but also build reputation and visibility. All this can be done during evenings, weekends or at any time outside your current "daytime" assignment as a student or employee at a company. Just be sure not to submit your letter of resignation until you are ready to give up your "day job."

When you start to look at opportunities, don't forget that globalization not only brings you global competition, but also global opportunities: One of the major benefits of open source is that all parts of the system are open to anybody. You don't have to look at what is needed in your local, say: Greater Zurich area, but you may find that there is a specific demand of a movie animation project based in New Zealand which exactly matches your interests. I also have seen posts in forums where people asked for specific career advice and got excellent answers. Don't forget: All this is free. You just need to go out there and find one of the thousands of exciting opportunities.

Once you feel you have found the right opportunity, then free up some of your professional time. Consider transitioning your job to part-time or to a freelancer or contractor role. A perfect way to get hired by a company like db4objects is to propose a part-time project, which adds real value, on a fixed price schedule. This shows initiative, self-confidence in the ability to deliver, self-manageability and many other positive attributes that open source projects and companies look for.

Keep in mind that open source communities have already become an important HR evaluation resource: db4objects and many other companies, including IBM, today systematically look at source code contributions and forum postings of their potential hires. More often than not, these companies hire the most valuable community members before they look at any other candidates. A long standing track record in relevant open source projects is often more valuable than a good grade in your university project<sup>9</sup>.

But there's more. It is not only important to write good software, but also to be able to effectively

communicate in and about your project. Consequently, it becomes important to refine your Internet-based communication skills. I personally have done an assessment of my professional and private life and systematically reduced many dependencies from non-Internet-based resources, while brushing up my e-mail, blogging, chatting, and posting skills. How efficient are you in communicating over the Internet? Can you extinguish a flame mail thread effectively? Can you keep remote friendships alive by e-mail, IM, and Skype? Believe it or not, you may find that your blogging skills result in higher pay at your next job!

Another piece of advice which I personally found helpful was to let go of perfectionism and to start embracing (and managing) the imperfect. Open source and Web 2.0, for that matter, live from the imperfect in massive parallelism. The only perfection that is really helpful is the perfection in being grateful and tolerant towards anybody's contribution, no matter how valuable it may appear to you at first sight. This doesn't mean that you don't know some or many things better yourself. But avoid knowing everything better yourself. I personally believe that every person has something to contribute, something that he or she can do better than me. If I cannot see it, I just haven't looked hard enough.

With the right habits and a traceable record of achievements in an open source project showcasing your skills, you will become a perfect hire for companies like db4objects and for companies who endorse the fourth element of open source: the global software development model based on contributions from distributed individuals.

## Links

1. <http://www.infoworld.com/articles/hn/xml/00/10/09/001009hnrs.html>
2. This change is driven by increased user adoption of these projects, so that the originators find themselves making their hobby project a viable livelihood. With increased adoption, open source projects become important strategic vehicles for some IT companies, so that they also dedicate resources to contribute to these projects. IBM's Linux Labs, for instance, has some 600 employees in 2004, to drive the adoption of Linux, Apache, and Eclipse, primarily to compete with Microsoft.
3. Netcraft August 2006 Web Server Survey [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)
4. <http://blogs.tedneward.com/>
5. A "seismic shift in the way value is delivered—through services, through middleware, through servers. We sell hardware under Linux, the software on top of Linux, and the services all around it. The fact we don't sell Linux itself is irrelevant in the big picture." IBM Annual Report, 2001
6. [http://news.com/Oracle+snags+open-source+database+company/2100-7344\\_3-6039070.html](http://news.com/Oracle+snags+open-source+database+company/2100-7344_3-6039070.html)
7. A detailed case study on MySQL's business model by Stanford Graduate School of Business' Prof. Robert A. Burgelman, Sami Inkinen and Christof Wittig can be found on Harvard Business Press Online (<http://tinyurl.com/lmqry>)
8. Tom Friedmann, "The World is Flat", 2005, ISBN 0374292884
9. Don't get me wrong: I am not saying that a university education is not useful, but a degree better shows in a competent contribution to an open source project rather than just as a grade on a piece of paper: "Non Scholae Sed Vitae Discimus" (We don't learn for the school but for life)
10. Robert A. Burgelman, Sami Inkinen, Christof Wittig, MySQL Open Source Database in 2004, (<http://tinyurl.com/lmqry>)

## Limits: More Products, Less Services

I should point out that the model described is particularly suited to software product rather than services, because service companies rely to a much higher extent on direct, physical customer interaction. This implies that the fourth element of open source will have higher impact to U.S. companies who are more license-revenue driven, versus service-revenue driven.

On the flipside, given the much lower cost to start a product company, we will see many more product companies with widely used off-the-shelf software rather than the abundance of small and mid-sized project- and service-oriented software shops, which are still so common in Switzerland and most parts of the world today.

Perhaps you are one of these entrepreneurs, too?

Today is a good day to get ready for the future!

△

## Comic

