Development, Ethical Trading, and Free Software

Final version November 30th 1999

Translations: en français (partial) | Serbo-Croatian

Abstract

This paper makes the political and ethical case for the adoption of free software by Community Aid Abroad and other members of Oxfam International. It should be applicable to development agencies generally and to other organisations with similar values.

Free software has obvious pragmatic advantages for community development processes, most notably in its empowerment of users. But the ideological foundations and social/organisational structure of the free software movement are also consistent with community development at a theoretical level.

Feedback on this paper would be appreciated: additional case studies would be particularly useful. A Community Aid Abroad appropriate information technology group has been set up, including a mailing list for discussions in this area. (This paper is also a subject of debate on Slashdot.)

Contents

- Introduction
- Software: Drawbacks and Dangers
- The Advantages of Free Software
- Free Software in Action
- Conclusion and Recommendations

Introduction

Informational goods make up a sizeable and increasing fraction of the world's trade - and an even larger fraction of profits, since margins tend to be higher. (Compare Microsoft's profit/turnover ratio with General Electrics'.) This trend towards an "information economy" is continuing. Ethical trading and appropriate technology policies should therefore cover informational products.

With some goods the major ethical concerns are in their manufacture or the effects on the environment of their use. Examples are wheat, iron, refrigerators, and so forth. Such goods are covered by a draft Oxfam GB Ethical Purchasing Policy, which advocates products that "are produced and delivered under conditions that do not involve the abuse or exploitation of any persons" and "have the least negative impact on the environment".

The policy mentioned considers weapons and baby milk powder as special cases. But there are many products other than weapons and baby milk powder whose production and delivery may raise no or only minor environmental and ethical concerns, but which may still have effects of major concern in the way they affect the autonomy and independence of users. It is the contention of this paper that software falls into this category.

This paper addresses only computer software. Other intellectual property issues are also of great importance. Control of genetic variability through gene patents is one example; World Intellectual Property Organisation treaties on copyright are another. (The latter ought to receive the same sort of critical response that the Multilateral Agreement on Investment did.)

> This is the context for intellectual property rights enforcement. This world *market* in knowledge is a major and profoundly anti-democratic new stage of capitalist development. The transformation of *knowledge* into property necessarily implies *secrecy*: common knowledge is no longer private. In this new and chilling stage, communication itself violates property rights. The WTO is transforming what was previously a universal resource of the human race - its collectively, historically and freely-developed knowledge of itself and nature - into a private and marketable force of production. (Allan Freeman, *Fixing up the world? GATT and the World Trade Organisation*)

A good deal of the world's primary resources are located in the poorer countries of the world's "South", even if their exploitation is often in the hands of external corporations. Systems for controlling the distribution of information, on the other hand, are (like possession of capital) overwhelmingly centralised in the rich "North". This should be of great concern to organisations such as Oxfam International members which take a long-term perspective in their attempts to reduce the inequitable distribution of resources. As the United Nations Sustainable Development Networking Program says:

> Information and Communication Technologies are now fundamental to dealing with all development issues in developing countries.

An Oxfam International Education Now report presents some of the consequences of an information economy for educational equity [DEAD http://www.caa.org.au/oxfam/advocacy/education/report/chapter1-3.html].

Software: Drawbacks and Dangers

The following analysis of potential political and ethical dangers in software is not meant to be complete. Nor is it an analysis which applies equally to all kinds of software. There are certain key components, such as operating systems, application programming interfaces, and software with mass deployment, on which many implementations and many other software systems depend. These are more critical than software systems with peripheral roles.

The Expense

Software is often prohibitively expensive. The standard price for an ordinary office package might be a year's income for most of the world's people. As one Mexican project adopting free software [DEAD http://www.linux.org.mx/arturo/scholar/] wrote:

> The primary reason for reaching this decision was the kind of money we would have had to pay if we went for proprietary software: at US$55 for each machine with Win98 and Office, US$500 for every NT license and an average of 6 workstations and one server for 140000 labs, that's a lot of money.

Though "discounts" are often available on software, these tend to either be in exchange for accepting a local monopoly for the vendor's products, or an attempt to gain market share at the expense of competitors. Consider, for example, Microsoft's attempts to bribe universities and colleges into using NT [DEAD http://207.178.22.52/articles/currents/007.html].

So called software "piracy" [DEAD http://www.corpwatch.org/corner/worldnews/twn/twn11.html] is obviously an option for those unable or unwilling to purchase software, and indeed it is a common choice throughout the South, where copyright law is often poorly enforced. But this places users at the mercy of the law, increasing their vulnerability to those rich and powerful enough to use it to their own advantage. Also, development organisations themselves are vulnerable to enforcement in their home countries, so they can not support or encourage such practices.

As well as the up-front costs of software, there are usually hidden costs. Often licensing is *per-user*, so costs will increase with the size of the user base and inhibit growth. *Support* for proprietary software is almost always prohibitively expensive. Frequent software *upgrades* may be required to maintain compatibility and functionality (consider the deliberate modification by Microsoft of the file format in successive versions of Word, in order to force users to upgrade to newer versions). And software tends, especially with upgrades, to require more powerful, and hence more expensive, *hardware*. These hidden costs are often recurrent.

Lack of Openness

Open standards and protocols are in the interests of consumers, and indeed of most businesses: they allow genuine market competition, giving users options and choices. Closed standards and protocols and technical secrets, on the contrary, benefit only those seeking to maintain or attain monopoly control of markets by decommoditizing software. (Proprietary software can, of course, use open standards and protocols, but much of it doesn't.)

Security and Privacy

The use of black-box proprietary software without source code creates security risks, since it makes the detection of Trojan horses rather difficult. One high-profile case is the Melissa Virus. An extreme case is government surveillance: an Australian government report (the Walsh Report, see sections 6.2.10 and 6.2.11) has recommended that security agencies arrange for back doors to be inserted into mass-market commercial software to allow eavesdropping. Perhaps it is paranoia to think that the United States National Security Agency has already arranged for this to be done, but when peoples' lives are at stake, can one *really* trust (say) Microsoft Word when vulnerable West Papuan or East Timorese activists are involved?

From a privacy point of view, some worrying features are *known* to have been built into popular proprietary software packages. Microsoft Windows and recent versions of Office include a unique computer identifier in all documents - an identifier which is sent to Microsoft on registration of software, as well as in cookies set by Microsoft's web site. The implications of this for anyone trying to maintain anonymous - whistle-blowers and activists most obviously - are frightening. (See analysis [DEAD http://byte.com/features/1999/03/win98priv1.html] and a news report from CNET; Paul Ferris points the obvious argument for free software in "Of Corporations, Privacy, and Open Source Software".)

The Creation of New Dependencies

Proprietary software increases the dependence of individuals, organisations, and communities on external forces - typically large corporations with a very poor track record on acting in the public interest. There are dependencies for support, installation and problem fixing, sometimes in critical systems. There are dependencies for upgrades and compatibility. There are dependencies when modification or extended functionality is required. And there are ongoing financial dependencies if licensing is recurrent.

Political dependencies can result from the use of proprietary software, too. For example, an Irish ISP under attack for hosting the top level East Timor domain .tp was helped out by hackers and community activists [DEAD http://www.zdnet.co.uk/news/1999/4/ns-6857.html] (setting up a secure GNU/Linux installation). Given that this attack was probably carried out with the connivance of elements of the Indonesian government, it is hard to see a commercial vendor with a significant market presence in Indonesia being so forthcoming with support.

Nearly exact parallels to this exist in agriculture [DEAD http://www.rafi.org/], where the patenting of seed varieties and genome sequences and the creation of non-seeding varieties [DEAD http://www.rafi.org/communique/19982.html] are used to impose long-term dependencies on farmers.

An Analogy: Baby-milk Powder

The effects of baby-milk powder on poor infants (which has sparked a Nestle campaign/boycott) provide an analogy to the effects of proprietary software.

Sending information in Microsoft Word format to correspondents in Eritrea is analagous to Nestle advertising baby milk powder to Indian mothers. It encourages the recipients to go down a path which is not in their best interests, and from which it is not easy for them to recover. The apparent benefits (the doctor recommended it; we will be able to read the documents sent to us) may be considerable and the initial costs involved (to stop breast-feeding and switch to milk powder; to start using Microsoft Office) may be subsidised, hidden, or zero (with "piracy"), but the long-term effects are to make the recipients dependent on expensive recurrent inputs, and to burden them with ultimately very high costs.

Moreover, because documents can be easily copied and because there are strong pressures to conform to group/majority standards in document formats, pushing individuals towards proprietary software and document formats can snowball to affect entire communities, not just the individuals initially involved.

Restrictions on Self-help

Proprietary software not only creates new dependencies: it actively hinders self-help, mutual aid, and community development.

- Users cannot freely share software with others in the community, or with other communities.
- The possibilities for building local support and maintainance systems are limited.
- Modification of software to fit local needs is not possible, leaving communities with software designed to meet the needs of wealthy Northern users and companies, which may not be appropriate for them.

An Example: Language Support

Language support provides a good example of the advantages of free software in allowing people to adapt products to their own ends and take control of their lives. Operating systems and word processing software support only a limited range of languages. Iceland, in order to help preserve its language, wants Icelandic support added to Microsoft Windows - and is even willing to pay for it. But without access to the source code - and the right to modify it - they are totally dependent on Microsoft's cooperation. (See an article in Seattle Times [was http://archives.seattletimes.nwsource.com/cgi-bin/texis.cgi/web/vortex/display?slug=icel&date=19980630] and an article by Martin Vermeer which argues that lack of software localisation is a threat to cultural diversity.)

Whatever the outcome of this particular case, it must be noted that Iceland is hardly a poor or uninfluential nation. There is absolutely no hope of Windows being modified to support Aymara or Lardil or other indigenous languages: the spread of such proprietary software will continue to contribute to their marginalisation.

In contrast, the source code to the GNU/Linux operating system is available and can be freely modified, so groups are able to add support for their languages. See, as an example, the KDE Internationalization Page [DEAD http://www.kde.org/i18n.html] (KDE is a desktop for GNU/Linux) or a project to localise GNU/Linux for Indian languages. Another example of the kind of thing that access to source code allows is the Omega Typesetting System [DEAD http://www.serg.cse.unsw.edu.au /DoSE/research.html], a modification of the free TeX typesetting system "designed for printing all of the world's languages, modern or ancient, common or rare"; this sort of extension or modification is simply not possible with proprietary word-processing packages.

Unsustainable

Sustainable development should favour unlimited resources over finite ones. But while software appears to be a renewable resource, its control by profit-making corporations, as "Intellectual Property", effectively turns it into a finite resource.

The Advantages of Free Software

What is Free Software?

The Free Software Foundation's "What is Free Software?" provides a good introduction to free software.

> `Free software'' refers to the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom:
>
> - The freedom to run the program, for any purpose.
> - The freedom to study how the program works and adapt it to your needs.
> - The freedom to redistribute copies so you can share with your neighbor.
> - The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

A key point is that "free" refers to liberty, not price. Software is only "free" if users have the freedom to copy, modify, and distribute it, and to share it with others. A key necessity for this is *access to the source code*.

Prominent examples of free software are the Linux kernel and the GNU system environment, which together constitute a complete operating system (an alternative to Windows or MacOS), and the Apache web server and sendmail mail transport agent, which between them provide more than 50% of the Internet's web sites and handle perhaps 80% of the world's email.

Pragmatic Advantages

Quite independently of any ethical and political considerations, there are also pragmatic and technical arguments for the deployment of free software systems.

- Some free software products are widely recognised as more reliable and robust, more powerful, and more secure than their proprietary counterparts, and a plausible argument can be made that this is not just accidental, but a *consequence* of their open development, implementation, and testing.

  Rob Bos puts it well in *32bitsonline* [DEAD http://www.32bitsonline.com/article.php3?file=issues/199902/abclinux11&page=1]

  > Free software is better than non-free software. It works better, it works faster, it works longer. Open source programs are tried and proven, they are constantly pressed from every direction to do specific tasks, and do them well; and for the simple reason that they are written to work, not simply to sell copies. Free software doesn't just work better, it works orders of magnitude better. Open sourcing an application gives the source code to a large number of developers, instead of a small, tight group. Free software projects have a pool of developers and an effective budget multiple times higher than an equivalent proprietary development project, and will, given all other equal things, advance at a rate many times faster because of their access to an much larger development team. Peer review of code isn't just a pipe dream, it is an essential means to writing superior applications, no matter where they are written.

- Free software can typically be obtained for the cost of the media (typically a few dollars for a CD) or network traffic (for distribution via computer networks). It can always be freely distributed. The pragmatic benefits of this should be obvious, but in some contexts the price of free software can also take on political significance:

> I live in India, one of the poorest countries in the world, with a large number of awfully bright, poor people. In India, today, the entry-level programmer (C knowledge but no work experience) earns $100 a month, and it is not routine for him to have a computer at home. Entry-level computers at $250 and below will attract millions of buyers in India, who will find the difference between $250 and $750 to be a massive one.
>
> Industry experts look at around 200 million existing computers, 80% of which run Microsoft OSes. It's useful to focus on the next billion computer sales. In this, I see the price-tag of $0 as being a critical product feature.
>
> Ajay Shah - Unix on a billion desktops? [DEAD http://freshmeat.net/news/1998/11/15/911138358.html]

- Free software is often less demanding of resources, extending the lifespan of older hardware. As an example of what this allows, consider **Project Computer Bank**, an Australian venture to supply old computer equipment running GNU/Linux to low income earners, community groups, and disadvantaged schools.

Freedom From Dependencies

> ``Community Aid Abroad's vision is for a fair world in which people *control their own lives*, their basic rights are achieved and the environment is sustained.'' (emphasis added)

Free software does not create dependencies on multinational corporations. Support commonly comes from user groups and online communities, which often provide vastly better support than commercial alternatives. Commercial support *is* available for free software systems, but users of free software can not be tied to single suppliers or vendors.

Access to the source code greatly increases users' options. It allows not just the unrestricted sharing of software packages but also their easy modification to suit local needs and requirements.

The value of free software in avoiding dependencies has been recognised by businesses and by governments.

> Let's say you are a chief technical officer (CTO) at a Fortune 500 company and you have just spent millions of dollars on a strategic business system with software you cannot see inside and cannot modify, software that depends on a single vendor to service. Now are those systems going to change to serve your business plan or your vendor's business plan?
>
> ...it probably will not be long before buying closed-source software for your key infrastructure is considered the height of irresponsibility.
>
> Eric S. Raymond in *Intellectual Capital* [DEAD

> http://www.intellectualcapital.com/issues/issue173/item1329.asp]
>
> "Scandinavia, Germany, and France are some of the main centers of Linux use. Some people say that this is because companies and the government want to avoid becoming too dependent on U.S. -- read Microsoft -- products."
>
> Kalle Dalheimer, quoted in OSS Europe [DEAD http://www.intraware.com/ms/mktg/indaa/itkc/osseurope.html]

Development of free software is done by those who have the necessary skills and resources - the resulting products are available for use by whoever needs it.

> With Linux, the people who use the system get to [affect the way] the system [develops]. It's democracy in the sense that you don't surrender control. Anybody can do anything. It boils down to [the fact that] you must be [competent], but that's a good way of separating the people who do the work. And even the [people who] don't make changes can make suggestions and can do testing and things like that.
>
> Linus Torvalds, interview with upsidedown.com [DEAD http://www.upside.com/texis/mvm/story?id=379de85d0].

Shared Values

Most free software has been produced through decentralised, community-based development processes which are usually open to anyone with the right technical skills (or a willingness to learn) who is prepared to do the work. Users of free software can join software development communities and participate in the refinement and improvement of existing software, or in the development of entirely new programs, building on what already exists.

Many free software development projects are almost model community development projects. They are based on open communication, inclusiveness, personal relationships, and working for the good of the community as a whole. In a paper Technology and Pleasure, Gisle Hannemyr describes the history of the "hacker" community, placing it in the artisan tradition and in opposition to Taylorism. He describes its imperatives as:

> reject hierarchies
> mistrust authority
> promote decentralization
> share information
> serve your community

and includes among its position statements:

> when creating computer artifacts, not only the observable

> results, but the craftsmanship in execution matters
> practice is superior to theory
> people should only be judged on merit (not by appearance, age, race or position)
> you can create art and beauty [NOWPAYTOVIEW http://www.theatlantic.com/unbound/digicult/dc990812.htm] by the means of a computer

The long-term effects of free software and associated changes are likely to be significant:

> Oscar Wilde says somewhere that the problem with socialism is that it takes up too many evenings. The problems with anarchism as a social system are also about transaction costs. But the digital revolution alters two aspects of political economy that have been otherwise invariant throughout human history. All software has zero marginal cost in the world of the Net, while the costs of social coordination have been so far reduced as to permit the rapid formation and dissolution of large-scale and highly diverse social groupings entirely without geographic limitation. Such fundamental change in the material circumstances of life necessarily produces equally fundamental changes in culture.
>
> Eben Moglen, *Anarchism Triumphant: Free Software and the Death of Copyright*

In many ways the ethics of free software reflect that of the Internet community more generally - a community which is still resisting commercialisation of the Net.

Education and Technology Transfer

Free software both encourages learning and experimentation and in turn benefits from it. Free software is widespread in educational institutions, since access to the source code makes free software an ideal tool for teaching: indeed much free software began as learning exercises.

Due to low start-up costs and rapid change, software development and the information economy more generally offer a possible way for the South to build high value industries, leapfrogging older technologies and even modes of production. The flourishing Indian software industry provides an obvious example. But if these industries are built on proprietary products and protocols owned by multinational corporations, then this will only reinforce one-sided dependencies. Free software has obvious advantages here.

Free software lends itself to collaborative, community-based development at all scales from cottage industry to world-wide efforts involving the collaboration of thousands of people. Internet access potentially offers the poor the ability to communicate directly with the rest of the world, to directly present their own ideas and perspectives. Combined with the free software development model, it allows them to participate in creating and moulding the technologies and systems that will determine their future.

Free Software in Action

The advantages of free software for community and development organisations have been recognised by others: The arguments sketched above apply not just to development organisations but to governments and to some extent even to businesses.

The United Nations
[DEAD http://www.unesco.org/events/latin/cd_linux_ing.html] UNESCO is handing out free Linux CDROMs to community, scientific, and educational projects in Latin America.

> We believe LINUX can play a very important role in Latin American and Caribbean modernisation, constructing networks to permit a great number of universities, colleges, schools and educational centers, to connect to Internet in order to use this fabulous tool to improve their scientific and cultural levels. In a few words, LINUX is the tool which permits to reduce the "technological gap" between the countries. LINUX permits the acces to "the informatics the most advanced" implemented according to the reduced economic capacities in our region. LINUX is a new way to make informatics, where the most important thing is "the technical quality and people solidarity"

And the UNDP is running a Sustainable Development Networking Program [DEAD http://www.sdnp.undp.org/home.html], with support from Red Hat and Corel [DEAD http://www.corel.com/international/southafrica/news/august_10b_1999.htm].

Mexico's Scholar Net
[DEAD http://www.linux.org.mx/arturo/scholar/]

> I work as the project leader of the "Scholar Net", a program that aims to bring computers and the net to every elementary and mid-level school in Mexico. We expect to install from 20 to 35 thousand labs per year to a total of 140,000 centers in the next five years.
>
> Due to matters of cost, reliability and configurability, we plan to use GNU/Linux to replace the proprietary server options and, now thanks to GNOME, the proprietary desktop application options.

SatelLife
SatelLife is an international not-for-profit organization employing satellite, telephone and radio networking technology to serve the

health communication and information needs of countries in the developing world.

> http://www.data.com/issue/981021/people.html [DEAD]
>
> For starters, the staff of Satellife had to seek out and master technologies cheap enough for users in the world's poorest countries but reliable enough to deliver vital medical information fast. And the organization didn't have the funds that corporate IT departments have for equipment and software-so it used free and open-source software to link users to forums. And as the Internet became a more vital tool, Satellife had to make sure that users without browsers could still get information via the Web. It also used second-hand gear where possible and relied on research institutes and discussion groups, rather than high-priced consultants, for advice.

The Littlefish Health Project DEAD http://www.paninfo.com.au/intro/littlefishproject_homepage.htm

> **Project Vision:** "To create a user friendly patient information and recall system on an open source basis with the focus on use by community based primary health care health organisations in the developing world or remote and rural areas or areas of need.

(See also a newspaper story on Littlefish [DEAD http://www.it.fairfax.com.au/hardware/19990607/A44136-1999Jun7.html], and a paper by Daniel L. Johnson on DEAD http://lorenzo.uwstout.edu/QQMIM/medicalfreesource.htmlfree software in medical information management.)

The Community Broadcasting Association of Australia (CBAA)
An impassioned but carefully put together argument for use of GNU/Linux by the CBAA [DEAD http://www.physics.usyd.edu.au/~matthewa/cbaa/cbaa-server.html]. Arguing on both technical and ethical grounds, most of this is directly relevant to development organisations.

> The open source movement, and Linux is particular, are massive volunteer non-profit projects which share the spirit of community media. It's a radical alternative movement creating successful mainstream software. In fact, it's the same movement that produced the software that the internet revolution depends on. Now the movement has produced a cutting edge technology which *suits the CBAA's needs far better than the commercial competition*. The technology is Linux. A Linux server is one the CBAA could be proud of.

(Also mentioned in a Newswire story on the politics of software [DEAD http://newswire.com.au/9902/polit.htm].)

Conclusion and Recommendations

The free software movement embodies principles consistent with those of Community Aid Abroad and Oxfam International. Free software products are tools which fit the needs of Oxfam International members, in many cases better than alternative proprietary products.

It is therefore recommended that:

- Development organisations should include software in their policies on ethical purchasing and appropriate technology; such policies should encourage the use of free software.
- Development organisations should encourage and assist project partners in the deployment of software systems that will enable them to "take control of their own destiny" and to reduce their dependence on the developed world. They should consider the major advantages free software has in this area.
- Development organisations should ultimately try to free themselves from the shackles of proprietary software.

Danny Yee <danny@anatomy.usyd.edu.au>

The author is one of the Community Aid Abroad webmasters, a board member of Electronic Frontiers Australia, and an employee of Sydney University. But the opinions expressed in this paper are personal and do not necessarily reflect the policies of any of these organisations.

Thanks to Cameron Tampion, Mike Gifford, Charlie Brady, Greg Taylor, Ronni Martin, and Richard Stallman for feedback on this document.

Free Software Advocacy << Danny Yee