Advantages and Disadvantages of Changing to Open Source Software

Anthony Glenn

2005 March 1

About Me

- Member of PCUG since 1984
  - Started word processing in 1984 with Word for DOS on my XT clone.
  - Used Word for many years, DOS and Windows
- Ordinary user, just like you
- Became unhappy with proprietary software.
- Amazing progress in Open Source Software
  - Loaded Open Office in 2003, when V1.0 appeared.
  - Loaded Mozilla and Thunderbird in 2004.
- Learning Debian distribution of Linux.
- Transitioning away from proprietary software al-together. Never going back.

I know what you are thinking.

- Why should I change to OSS?
- What about the years I have spent learning Windows and all my applications?
- You devil, you want me to lose all my documents, pictures and other data files. Don't you?
- Aren't the OSS applications immature?
- Isn't Linux too nerdy and just plain scary?
- I am too old, too busy, too fixed in my ways. You can't make me change!

Monsters Coming To Get You

- High prices
- Consumer-hostile license conditions
- Legal threats
- Privacy threats
- Threats to your data
- Poor reliability
- Consumer-hostile technology
- Your freedom is being taken away
- Foolish legislators
- Forced harm to your neighbours
- Unresponsive vendors
- Lack of support

Prices

- Windows XP Home, Full Version, $324
  - No way to get some missing features, other than by paying a lot more for XP Professional.
  - CD, small book and a cardboard box
  - Also need firewall, anti-virus, anti-spyware, etc., $100
- Debian Woody 3.0 for x86, DVD-R, $25
  - Most Linux home distributions are less than $50.
  - Many distros available for free download.
  - Your friends can legally give you a copy.
- Office 2003, $699
  - Pay that price for every computer you install on.
- Open Office, Firefox, Thunderbird, all $0
  - Free legal downloads from OpenOffice.org, Mozilla.org
  - Legally install on as many computers as you like.
- Consultants Prefer OSS
  - Customers often have illegal stuff, want things for free.

Prices

- High prices for proprietary software
  - Permanent situation
  - Vendors will always charge what the market will bear, regardless of cost. Harsh facts of commerce.
  - Vendors trying hard to lock out all competition. Examples, DR-DOS, Word Perfect. If they succeed, watch out!
  - Barriers to entry rising all the time. Foolish legislators passing more and more bad laws. Entry-level product getting more complicated. Example, word processing.
- Low prices for OSS
  - Permanent competition from other OSS vendors
  - Permanently legal to give your friends a copy
  - Low barrier to entry, guaranteed by GPL
  - Economic doctrine of “commodification of the complements”
  - Revenge of outraged consumers

License Conditions

- Proprietary Software License
  - Install on only one computer
  - Problems likely with multi-core processors
  - Takes away your right to deal with your property as you see fit. Compare with the situation when you buy an appliance.
  - Prohibition of reverse engineering, disassembly
  - Forced agreement to product activation
  - “Free” upgrades may force further license conditions
  - Vendor has right to cancel your license any time they like
  - No right to sell license, forces destruction of software when computer changes hands. Example, bankrupt sale.
- OSS General Public License
  - Install on as many computers as you like. Multiple cores OK.
  - Modify freely, understand freely
  - No product activation, no upgrade conditions
  - Permanent right to use the software
  - When computer is sold, software can legally go with it
  - Give away copies to your friends, customers

Legal Threats

- Licenses are legally enforceable.
  - Proprietary vendors can take you to court for any license infringement, cost you big money.
- The Business Software Association of Australia (BSAA) exists to legally harass software users.
  - Offers cash rewards for information leading to a conviction
  - Wise consumers stay away from BSAA members.
- Awesome powers are granted to a copyright owner by an Anton Pillar order.
- Software audit can be imposed on you.
- Patent infringement can be alleged.
- Coming to Australia Soon
  - Digital Millennium Copyright Act (DMCA), will make mod chips illegal.
  - Software Patents a huge threat to software developers.

Privacy Threats

- Software tries to “Phone Home”
  - Why?
  - Typically just after it has been installed
  - Firewall reports attempt, many users allow attempt
  - Can be forced on you by product activation
- Spyware
  - Reports anything “they” want to know
  - Can allow identity theft, fraud
  - Banking details a particular target
- Viruses, worms, adware, back doors
  - This stuff and spyware go together, help each other
  - Often tries to disable anti-spyware software
  - Waste your computer resources, cause poor performance
- Not Tolerated by OSS Community
  - Community is young, idealistic, alert
  - OSS and “many eyeballs” effect guarantee any bad stuff will be found fast, fixed fast. Bad guys do not even try.

Threats to Your Data

- Proprietary Formats
  - Deliberately poorly documented, as a competitive weapon.
  - Your data gets automatically converted into the latest format.
  - Keeping your data in any older/foreign format is deliberately made inconvenient, annoying.
  - Formats from several versions ago are deliberately not supported, to try to force upgrade revenue.
  - If you fail to convert when you have the chance, the next version of the software might not let you convert. You lose your data.
  - Encrypted formats coming. Already standard in game consoles.
  - Patented formats coming. Will prevent OSS from reading the proprietary format at all.
  - Digital signing coming. Software developers have to sign a restrictive license agreement, pay money. OSS developers will refuse.

Threats to Your Data

- Proprietary Formats
  - Deliberate failure to conform to standards means that when you try to use different software, things go wrong. That is an interoperability problem.
  - The purpose is to try to force customer lock-in.
  - Interoperability problems can effectively destroy your data and lock you into using a particular item of proprietary software.
  - Also stop you from using utilities from anybody other than the proprietary software vendor.
  - For example, you might be particularly interested in a utility which checks for standards compliance.
  - Such a utility might be too inconvenient to use, if proprietary software insists on putting in non-standard things into your data. That effectively destroys the ability for you to check that your own data is standards compliant.

Threats to Your Data

- "Embrace and Extend" Explained
  - Proprietary vendor "embraces" some existing standard, could be a data format, API (Application Programmer Interface), protocol, etc.
  - Then they say, "We support Standard X." They get a tick on the corporate check lists.
  - Then they "extend", that is, add extra features, which are exclusive to them.
  - They make the extra features as proprietary as possible.
  - They fail to co-operate in open standards formation, patent stuff, copyright stuff, fail to publicly document, etc.
  - They use the extra features as heavily as possible.
  - Your data gets "infected" with the extra features if you use their proprietary software.
  - Then when you try to use other software, it does not work. You lose your data or suffer a data migration nightmare.
  - This technique resulted in a major argument between Sun and Microsoft over the Java programming language.

Threats to Your Data

- Convert now to an open format, before you get locked in!
  - Presently, OSS can do an excellent job of converting most proprietary formats. That will not last!
  - Time window of opportunity is closing. Proprietary software vendors want you to be locked in. They are working on it.
  - If you leave it too late, your data will be stuck in a proprietary format, readable only by proprietary software.
  - If that happens, stand by for high prices, large pain to get yourself out.
- Open Formats
  - Often developed by standards bodies, such as W3C (World Wide Web Consortium), OASIS (Organisation for the Advancement of Structured Information Standards) and many others.
  - Standards development is always slow and political. Cannot be helped.

Threats to Your Data

- Open Formats
  - OSS developers contribute heavily and willingly to standards development.OSS developers normally regard standards compliance as very important.
  - Then OSS software is made highly standards compliant.
  - Due to the beautiful standards compliance, all the OSS software interoperates nicely.
  - You do not lose your data.
  - You can change software without penalty.
  - The competitive marketplace guarantees low prices and high quality, permanently.
  - Alas, all the devotion to standards takes a long time and a lot of effort, but you, the customer, benefit.
  - Because standard compliance is hard, OSS gets beaten to the market by proprietary software. That is bad, but often unavoidable.
  - Customers need to be patient and tolerant. It will be worth it in the long run.

Poor Reliability

- Software Reliability
  - If the software you are using to modify your data crashes, then you are likely to lose your data.
  - OSS has a much better reliability record than proprietary software. Example, Netcraft uptime survey.
  - OSS developers are devoted and talented. They are often much more experienced and knowledgeable than their proprietary counterparts. They stay on the project for many years and really get to know it. Example, Andrew Tridgell.
  - Proprietary software vendors face continual staff turnover (caused by poor management). They end up with inexperienced developers. Poor quality software results.
  - Corrupting influence of upgrade revenue gives a motive to leave in or introduce bugs. Customers upgrade to try to get something which works properly.

Poor Reliability

- OSS Development gives Better Reliability
  - OSS allows someone who is not happy with the present direction of development to “fork” the software. Then they can develop their own version which competes with the original version. Example, Xfree86 vs X.org.
  - The OSS community decides which version is best, then the other one dies, due to lack of developers.
  - The threat of forking, keeps OSS developers focussed on quality and reliability.
  - Forking cannot happen with proprietary software, except by management action. Since management is usually the cause of developer unhappiness in the first place, management does not usually order a fork. Hence mistakes are preserved. The result is diminishing reliability. Example, Macintosh OS 9 vs OS 10.

Poor Reliability

- OSS Development gives Better Reliability
  - The GPL allows anybody to further develop the software at negligible cost of starting development.
  - The OSS community strongly encourages anybody who has made useful developments to submit them to the “main line” and have their developments part of the standard software.
  - Exactly that happens, on a large scale.
  - Worldwide, every university computer science department which teaches operating systems (which is a lot of them) is using Linux, which is the **only** operating system for which the full source code is readily available. Likewise, all the various Linux-loving companies have people looking at code. So Linux gets the benefit of the massive brain power of all those "researchers" in many universites and many companies.
  - So faults get found **before** the bad guys have a chance to do something nasty. Then they get fixed so fast that the bad guys do not bother **trying** to do anything nasty. So you, the user, just never see a problem.

Poor Reliability

- Poor company culture degrades reliability
  - Famous “Don't tell me your problems, just get it done!”
  - Short term hires.Management expects people to be productive fast, declines to provide adequate training. Result: developers who do not really know what they are doing.
  - Developers feel, “It's just a job.” Lack of deep knowledge and passion.
  - Lack of adequate testing. Testing is a source of bad news. Dumb managers neglect testing, leaving defects undiscovered and unfixed. Customers find out the hard way later, by suffering poor reliability.
  - Management allows “crunches” as the shipping date approaches. Developers work far longer than normal working hours per week. Exhausted people make mistakes.

Consumer-Hostile Technology

- What is it?
  - Any deliberately inserted product feature which prevents consumers from doing what they want with the product.
  - Not the same as bugs, mistakes, omissions or other unintentional defects, this is deliberate.
  - Usually not advertised, you find out the hard way, when you have already paid for the product.
  - It is an ambush attack against the consumer, usually done to increase revenue.
- CHT Examples
  - Regional coding on DVDs and DVD players, prevents consumers from buying from possibly cheaper, better stocked suppliers outside the region. Restricts consumer choice.
  - Regional coding now exists on game consoles.
  - Regional coding will be here for software when the vendors figure out how to make it stick.

Consumer-Hostile Technology

- More CHT Examples
  - CSS (Content Scrambling System) on DVDs, prevents “unauthorised” players of DVDs. System is rigidly enforced by legal threats from the entertainment industry.
  - Macrovision copy prevention system prevents you just playing a DVD into a VCR (Video Cassette Recorder), thereby making a copy.
  - Disabled transport controls on DVD players, prevent you fast-forwarding past annoying legal messages or advertisements.

Consumer-Hostile Technology

- More CHT Examples
  - Product Activation, prevents you from reinstalling as you please, forces compulsory registration.
  - Limited time, limited run attempts, prevent you from using software without forced “upgrades”.
  - Install software which will only allow upgrades, not a complete fresh install. Makes recovering from a hard disk crash needlessly difficult.
  - Digital Rights Management, designed to prevent you from copying freely.
  - Not providing documentation, overcharging for it, or insisting on a license agreement before releasing documentation. Used as a competitive weapon against other developers. Results in reduced consumer choice, higher prices.
  - Same goes for deliberately leaving in errors in documentation.

Consumer-Hostile Technology

- More CHT Examples
  - Encrypted file formats, with legal threats against anybody who does “unauthorised” decryption. Example, Ebook, DVD CSS.
  - Attempts to sabotage standards formation, by causing trouble in the standards committee.
  - License conditions forbidding reverse engineering. Example, Samba project dares not disassemble any Windows code. Everything must be done by inspecting bytes on the network.
  - Threats of legal action against other developers.
  - Arbitrary changes directed at competing developers, change APIs, fail to deliver promised features. Example, Word Perfect's problems under Windows.
  - Deliberate failure to conform to standards. Makes other developers have to develop special techniques to cope with the non-standardisation. Example, Internet Explorer, Command sets on remote controls.

Consumer-Hostile Technology

- More CHT Examples
  - Persuading foolish legislators to pass laws contrary to the best interests of consumers.
  - Using already passed laws against consumers.
  - Insisting that device drivers be “digitally signed”, then refusing to co-operate with certain developers who are deemed unpopular.
  - Abuse of patent system.
  - Failure to support other file formats. Results in customers having difficulty changing software.
  - Providing an import filter for some competing file format, but no matching export filter. Result: transferring your data is a one-way trip.
  - Printer cartridges made with a needless chip in them designed to make the cartridge run out early and not be capable of being refilled.

Consumer-Hostile Technology

- More CHT Examples
  - So-called “trusted” computing is coming. Also known as “treacherous” computing, by consumer advocates. The owners of content will control what you can and cannot do with things like music files on your own computer.
  - You are not allowed to have a bug fix unless you can identify yourself as a legal owner first. Say goodbye to freely available upgrades, for proprietary software. Meanwhile, OSS will still have free upgrades, because the whole software is free.

Consumer-Hostile Technology

- I am not making this up. All this stuff is really happening.
  - A vast amount of work has been done, inventing CHT. More may be expected.
  - Foolish legislators and lazy public servants are not doing enough to protect consumers.
  - Blind, deaf, reluctant-to-bite watchdog syndrome.
  - Your only realistic defence is to have a consumer backlash against CHT.
  - Get aware, get informed, get angry. Blacklist the companies which support CHT.
  - A consumer backlash is the only thing the companies cannot ignore, but it has to be big enough to hurt them.
  - When you buy a printer, ask your local cartridge refiller for a recommendation, before you buy.
  - Change to OSS.

Freedom Taken Away

- Would you buy a car with the hood welded shut?
  - You want your car to be capable of being fixed by you or your favourite local mechanic.
  - You would be outraged if you were prevented from doing your own car maintenance, should you want to.
  - Expect to be overcharged if your mechanic cannot compete for your business.
  - There is actually is a little bit of the car which is effectively welded shut – the car computer.
  - Who owns this car anyway? You should be able to deal with your own property as you see fit.
  - It is time to take the same attitude to software.
  - Insist on having your freedom.
  - Take back your rights.

Foolish Legislators

- Proprietary software companies are always trying to increase their rights under the law and decrease consumer rights.
  - Vast sums are spent on lobbying American legislators.
  - American laws keep getting changed to benefit wealthy companies.
  - Legislators regard it as acceptable to criminalise behaviour engaged in by a majority of the population.
  - Whole problem is just getting bigger and bigger, tracked under heading “Your Rights Online” on Slashdot.
- Recent legislative changes in USA include:
  - Further extension of copyright terms, harmful to USA culture.
  - The DMCA (Digital Millenium Copyright Act), criminalises any attempt to defeat any form of anti-copying technology.
  - Software patents, allow a vast number of trivial ideas to be patented, thereby laying a huge legal minefield for anybody developing software.

Foolish Legislators

- Recent legislative changes in USA include:
  - USA is going to have a “broadcast flag” on digital free-to-air television, thereby outlawing home taping and time shifting.
- Patent system is broken worldwide
  - Patent offices are all too stupid to create a unified system worldwide. They all issue their own patents and duplicate a vast amount of work. Result is costs so high that only big companies may play. Everybody else faces a legal minefield.
  - More and more trivial things are getting patented. The minefield is getting worse.
  - Patent terms are getting longer, slowing down innovation
  - Useful website: www.nosoftwarepatents.com

Foolish Legislators

- Our local Aussie legislators and public servants are total “babes in the woods”.
  - They have signed a “free trade agreement” which “harmonises” the intellectual property laws in USA with Australia.
  - USA companies just have to persuade their own foolish legislators and ours will automatically pass similar legislation.
  - Saves them money, costs us our money and our rights.
  - Our legislators will not listen to consumers. The money from the big companies is talking.
  - The only effective sanction is to adopt OSS.

Forced Harm to Neighbours

- Proprietary software makes you be nasty to your neighbours.
  - “Neighbour” as in “Love thy neighbour as thyself.”
  - If you have spent time learning some software, it is easy to get enthusiastic about it.
  - You naturally want your neighbours to enjoy the benefit as well.
  - So you give them a copy -- all nice and legal with OSS.
  - But with proprietary software you are legally forced to be not neighbourly. All you can legally do is suggest they go and spend their money.

Forced Harm to Neighbours

- Sending people a file in a proprietary format.
  - Not a neighbourly thing to do.
  - Common mistake, caused by inertia.
  - Recipient has to find software which understands the proprietary format. That is getting more difficult and expensive.
  - Or, they have to ask you to send it again in a standards compliant (open) format.
  - Be nice to your neighbours, stick with standards compliant formats from the beginning.
  - OSS always supports standards. OSS developers have the attitude, “We will be standards compliant or bust!”
  - Proprietary software supports standards too, but reluctantly, badly and late.
  - Alas, you need to know which formats are standards compliant and which ones are proprietary. There is no substitute for knowledge.

Unresponsive Vendors

- Proprietary software vendors are in it for the money.
  - If you want them to take out Consumer-Hostile Technology, they do not want to.
  - If you want bugs fixed, they do not want to even admit that bugs exist.
  - Watch out for vendors who talk about “issues” not “bugs”.Saying “bugs” is honest. Saying “issues” is not.
  - Their reliability is poor, so they are flooded with complaints.
  - Companies routinely under-resource customer support. You get stuck on the phone until you give up.
  - You have to pay for major upgrades. So lots of people are using older versions. Then support for the older version gets withdrawn.

Responsive Vendors

- OSS vendors are in it for the community.
  - Consumer-Hostile Technology is not there.
  - If you want bugs fixed, OSS has comprehensive bug reporting and tracking systems, usually publicly available over the web. Example, Debian Linux distribution.
  - Upgrades are free. So it will be expected that you are running the latest version. Make sure you know how to do upgrades.
  - OSS reliability is excellent, so your complaint will get attention, provided it is genuine.
  - OSS developers have a low tolerance of people who have not made a genuine effort to help themselves first. You do not get the benefit of the doubt. You need to do your homework.
  - If you want hand-holding services, there are many consultants who will be happy to help, for a fee.
  - With OSS, support is “contestable”. Barriers to entry are low. Anybody who wants to can set up a consulting business and provide hand-holding services.

Software Should be Free

- “Free” means libre here. Why should it be free?
  - Software is really the instructions for making a machine. It happens to be a machine made of bytes that has to run in a particular environment, but it is still a machine.
  - Every living creature is also a machine and its software is its genetic code.
  - Machines are incredibly long-lived and numerous.
  - No matter how vast the cost of development was, due to the vast numbers and long life, the cost of development is insignificant per machine.
  - The only costs which matter are (1) the cost of copying the instructions, (2) building the machine.
  - For living things, (1) is tiny, but (2) is large.
  - For software, both are insignificant. (1) is paid by the software vendor. (2) is paid by the customer.
  - Therefore, software should be available for the cost of duplication, that is, very cheap.

Software Should be Free

- Why should it be free?
  - Having someone charge you hundreds of dollars for a CD, a book and a cardboard box is an outrage.
  - As a rational consumer, you can and should reject outrageous prices.
- How can the development get paid for?
  - The huge advantage of OSS is that anybody can get, something which can be further developed, at negligible cost. The GPL is specifically written to allow that.
  - The barrier to entry is low.
  - So if you want some special feature just for you, you pay only for the development of that feature.
  - Very often, people who want some feature also want general improvements elsewhere, and are willing to pay for the general improvements.

Software Should be Free

- How can the development get paid for?
  - Then the general improvements get made available to everybody. There is elaborate machinery set up by the OSS community to allow that to happen.
  - So lots of people, pursuing their own selfish ends, end up contributing to the community good.
  - The whole community shares in the goodness by not having to pay high prices for software and by having the freedom to develop it themselves.
  - The rejection of high prices is the revenge of the consumers.
  - The freedom to develop is the revenge of the programmers.
  - The killer strength of OSS is the strength of the community.
- Proprietary software does not get a community.
  - Everybody knows that any improvements will show up in even higher prices for the product. Example, Windows and Office.
  - Nobody wants to join a community where somebody else gets the benefits, so they don't.

Software Should be Free

- Results
  - Proprietary software companies have to pay for everything they get. They can never compete for quantity and quality of developers, except in niche markets.
  - OSS keeps on getting better and better for a cheap price.
  - Proprietary software cannot keep up because they keep on losing customers. Their revenue problems gradually get worse and worse.
  - All the hardware manufacturers are supporting Linux, because they want software to be a commodity.
  - It is rapidly getting to be the whole computer industry against a tiny group of proprietary software companies.
  - The forces arrayed against the proprietary software companies are large and are rapidly becoming overwhelming.

Software Should be Free

- Conclusion
  - In any large market, proprietary software is doomed. OSS just gets relentlessly better and eats away the customer base.
  - In niche markets, a company might be able to own all the developers and find customers desperate enough to pay high prices. So proprietary software will hang on there.
  - If you are a member of a large software consumer group, you are entitled to low prices and high quality.
  - Accept no less. Get nasty if anybody tries to take it away from you.

Software Should be Free

- Chicken Analogy with OSS
  - This chicken analogy actually applies to all living things. Hey, I like talking about chickens.
  - Every chicken inherits its genetic code from its ancestors, at no cost.
  - Many chickens have died to develop the chicken genetic code. Many things have been tried, with mutations. Only a few mutations have been successful. The cost to chickens of the code development has been immense.
  - But since there are so many chickens, the cost per chicken is negligible.
  - Any chicken may attempt further code development, by having a mutation, or just standard genetic mixing.
  - Natural selection will then sort out the successful code.
  - As time goes by, chickens enjoy the benefits of low cost, reliable, genetic code.
  - Most chickens are extremely good at being chickens, so this technique works.

Conclusion

- The era of OSS has started.
- That is the final stage in software.
  - Proprietary software will retreat to niches.
- There is no point in persevering with doomed software.
  - They can never make it reliable enough.
  - They cannot fix it, it is too broken.
  - It will be killed by a rising tide of malware.
  - Their development will hit a wall of complexity.
  - Customers will rebel against consumer-hostile technology.
- Change to OSS now.
  - Yes, it will be painful learning new stuff.
  - Support for unusual hardware devices is spotty, now. That will improve.
  - There will be no big company to call.
  - Call your local friendly computer consultant instead.
  - At last, your computer will “just work”.