**What is it?**

Open source software is generally free software that you can use in your business. Open source developers choose to make the source code of their software publicly available for the good of the community and  to publish their software with an open source license – meaning that other developers can see how it works and add to it. Examples of open source products include Open Office, the internet browser [Mozilla Firefox](http://en.wikipedia.org/wiki/Mozilla_Firefox), Wikipedia, the [GNU/Linux](http://en.wikipedia.org/wiki/Linux) operating system and its derivative [Android](http://en.wikipedia.org/wiki/Android_(operating_system)), an operating system for mobile devices.

**How does it work?**

From a business user perspective, open source software works in much the same way as proprietary software systems provided by commercial software firms – the only difference being that generally you don’t pay for it. However there are a few important differences – the idea behind open source software is that users are effectively co-developers, suggesting ways to improve it and helping to hunt out bugs and problems. This means that if you wish, you can modify it to your own needs, port it to new operating systems and share it with others.

**What are the advantages of open source software?**

1. It’s generally free – it has been estimated that open source software collectively saves businesses $60 billion a year. These days for virtually every paid for proprietary software system you will find an open source version.

2. It’s continually evolving in real time as developers add to it and modify it, which means it can be better quality and more secure and less prone to bugs than proprietary systems, because it has so many users poring over it and weeding out problems.

3. Using open source software also means you are not locked in to using a particular vendor’s system that only work with their other systems.

4. You can modify and adapt open source software for your own business requirements, something that is not possible with proprietary systems.

**Any disadvantages?**

1. Because there is no requirement to create a commercial product that will sell and generate money, open source software can tend to evolve more in line with developers’ wishes than the needs of the end user.

2. For the same reason, they can be less “user-friendly” and not as easy to use because less attention is paid to developing the user interface.

3. There may also be less support available for when things go wrong – open source software tends to rely on its community of users to respond to and fix problems.

4. Although the open source software itself is mostly free, there may still be some indirect costs involved, such as paying for external support.

5. Although having an open system means that there are many people identifying bugs and fixing them, it also means that malicious users can potentially view it and exploit any vulnerabilities.

**The practicalities**

You can download open source software onto your computer system in the same way you would proprietary software. Some software providers such as Alfresco, MySQL and Ingres offer both open source versions of their software and paid-for proprietary versions.

**Things to consider**

Because of the way it has been developed, open source software can require more technical know-how than commercial proprietary systems, so you may need to put time and effort into training employees to the level required to use it.

**Top tip**

Start with the most popular open source software systems that have built up a large community of support behind them, so you have somewhere to go to if you need advice.

**Where can I find out more?**

Useful websites include [Opensourcewindows.org](http://opensourcewindows.org/)and [Opensource.org](http://opensource.org/).

We also talk about this in our post about [the internet of things](http://www.entrepreneurhandbook.co.uk/the-internet-of-things/), check it out!

# Benefits of Using Open Source Software

The outline flow of this section is as follows:

* Organisational benefits from the use of Open Source Software
  + [Reliability](http://open-source.gbdirect.co.uk/migration/benefit.html#reliability)
  + [Stability](http://open-source.gbdirect.co.uk/migration/benefit.html#stability)
  + [Auditability](http://open-source.gbdirect.co.uk/migration/benefit.html#auditability)
  + [Cost](http://open-source.gbdirect.co.uk/migration/benefit.html#cost)
  + [Flexibility and Freedom](http://open-source.gbdirect.co.uk/migration/benefit.html#flexibilityfreedom)
  + [Support and Accountability](http://open-source.gbdirect.co.uk/migration/benefit.html#supportaccount)

## Introduction

Open Source's proponents often claim that it offers significant benefits when compared to typical commercial products. Commercial products typically favour visible features (giving marketing advantage) over harder-to measure qualities such as stability, security and similar less glamorous attributes. As a shorthand, we shall describe this phenomenon as **quality vs features**.

Open Source Software developers are evidently motivated by many factors but favouring features over quality is not noticeable amongst them. For many developers, peer review and acclaim is important, so it's likely that they will prefer to build software that is admired by their peers. Highly prized factors are clean design, reliability and maintainability, with adherence to standards and shared community values preeminent.

"The Open Source community attracts very bright, very motivated developers, who although frequently unpaid, are often very disciplined. In addition, these developers are not part of corporate cultures where the best route to large salaries is to move into management, hence some Open Source developers are amongst the most experienced in the industry. In addition all users of Open Source products have access to the source code and debugging tools, and hence often suggest both bug fixes and enhancements as actual changes to the source code. Consequently the quality of software produced by the Open Source community sometimes exceeds that produced by purely commercial organisations." ([QINETIQ2001](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#qinetiq2001)).

"... This psychological effect is really important in explaining why so many projects are started out of the blue, with seemingly no reward. Although it may seem surprising at first view, it is not that rare if we put it in context. For instance, most of the history of information science and programming, in fact, started this way in academic circles. And still many non-applied sciences advance thanks to the work of scientists who feel more rewarded by research in itself than by money. Although this effect of self-reward is perhaps not so common in the world of proprietary software development, it is today a strong force in the open source community. And what is even more important, it seems clear that it has an extremely good impact on developer's productivity, an interesting effect in a discipline where differences in productivity from person to person are often a matter of orders of magnitude." ([CONECTA2000](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#conectapaper)).

There is abundant anecdotal evidence for truth in those claims but reliable and comparable statistical measures are not available to our knowledge. There are a number of principal reasons adduced for this:

* Authors are motivated by pride and peer recognition rather than a development plan supplied by the marketing department. Most want to use the software themselves and they prefer robustnesss before adding features.
* Authors are likely to consider it a ‘win’ if they can reduce the complexity and improve the maintainability of software. This rarely comes high on the product plan for commercial software.
* Where several authors work in parallel, the best-of-crop solution can be chosen in place of the only solution (as would be typical for a commercial product).
* Where source code is freely published and widely distributed, the users of the product will often discover and correct defects themselves. If no commercial entity benefits from that work, the motivation to do so for the common good is **much** higher.

Many of the qualities that are claimed for Open Source Software are exactly those that are wanted by those who have to **use or deploy** software, yet run almost directly counter to the commercial needs of typical software **development** businesses, where a continuous revenue stream (usually through the mechanism of upgrades or high-priced support) is needed.[[CONECTA2000]](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#conectapaper) addresses some of the economic drivers in more depth and provides models which can reverse this apparent conflict of interests.

In the following paragraphs we look at the claims, outline why they are considered benefits and describe the ways that the Open Source process provides substance to the claims.

## Reliability

Reliability is a loose term. Broadly, we can take it to mean the absence of defects which cause incorrect operation, data loss or sudden failures, perhaps what many people would mean when they use the term `bug'. Strictly, a bug would also mean failure to meet the specification, but since most Open Source projects dispense with the concept of anything easily recognisable as a formal specification, it's hard to point to that as good way of defining what is a bug and what is a feature. Determining what constitutes a bug is usually by agreement amongst the developers and users of the software (an overlapping community in many cases). Obvious failure to perform is easily recognised as a bug, as is failure to conform to appropriate published standards. Security related failings (exploits or vulnerabilities) are clearly bugs too. Each of these kinds of bugs is usually addressed with speedy fixes wherever possible and Open Source advocates will claim very rapid time-to-fix characteristics for software.

Severe defects tend to be fixed within hours of their being detected, a process wich is undoubtedly assisted by the availability of the source code. Able developers who discover a bug will commonly also fix it and then report it to the maintainers as well as issuing an updated version of the software on their own authority. Users of the software can choose whether to use the unofficial fix or wait for an `official' version. By `official' we mean a release blessed by the project team itself or a trusted authority such as one of the main distributors of Open Source packages. This mechanism clearly works very well in practice.

The pattern with closed-source software is typically that a defect report needs to be filed and then there will be a delay before the vendor determines when or whether to issue an updated release. Users of the software are much more at the mercy of the vendor's internal processes than with the Open Source arrangement and the personal experience of the authors is that it can be extremely frustrating to move from the Open Source to the closed model.

"The market greatly values robustness, and the Open Source model, particularly as practiced by Linux, encourages a large market of early adopters (compared to the size of the early market for commercial products) who actively help debug the software. Consequently much Open Source software becomes highly robust at a surprisingly early stage of its development, and mature Open Source products are setting new industry standards for bulletproofness." ([QINETIQ2001](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#qinetiq2001))

## Stability

In a business environment software is mostly a necessary evil, a tool to do a job. Unless the job changes or more efficient processes are discovered then there is rarely pressure or need to alter the software that is being used to assist the task. This is more or less directly counter to what motivates software vendors who are in the unenviable position of supplying a commodity that does not wear out or age much. The vendors need a stable revenue stream to be able to keep their business going whilst their customers have not the slightest desire to change or upgrade any product that is working well enough to suit their needs. If a software supplier can establish a virtual monopoly and then force upgrades onto its audience (as has been the history of the software industry since the mid 1960s) then the profits can be very high.

Software vendors can apply a number of tactics to persuade their customers to upgrade more or less willingly. Typical tactics include moving to allegedly new and improved file formats (which require the new and improved software to read them) or to withdraw support and bug fixes for older versions after a short period. The problem for users of the software is that they rarely have much control over that process and are left isolated if they choose to remain with older versions that they consider to be acceptable. This has cost and control implications for the business.

Open Source Software is not a panacea in the world of ever-changing software, but the worst effects of vendor-push can be mitigated. The way that Open Source products tend to conform closely to standards efforts has an inertial effect, since standards change but slowly and interchange formats are often particularly stable. As a result, incompatible file formats can be less of an issue. If they are standards-based then they typically aren't an issue at all, and if they are formats unique to the software product — proprietary formats in a sense - then they cannot be undocumented since the source code that uses them is itself published. In practice the track record of Open Source projects is usually good; when incompatible formats are used it is commonplace for a Perl or similar converter program to be shipped with them which will upgrade data to the new format.

In the real world, no business is static and software changes to meet new requirements. A choice to use Open Source software can provide a counter to the pressures to upgrade for the vendor's commercial purposes but cannot shelter every user from any change. Having access to the source code can allow a business to choose to support itself on an old version where necessary and we belive that in general it gives more options and choice to the users. Nonetheless, some upgrading and maintenance effort will always be needed. Putting the choice in the hands of the users rather than the suppliers is hard to criticise.

## Auditability

A rarely-understood benefit of Open Source software (any software where the source code is published) is its auditability. Closed-source software forces its users to trust the vendor when claims are made for qualities such as security, freedom from backdoors, adherence to standards and flexibility in the face of future changes. If the source code is not available those claims remain simply claims.

By publishing the source code, authors make it possible for users of the software to have confidence that there is a basis for those claims. Whether this takes the form of an cursory and informal inspection or more rigorous auditing, what's clear is that without access to the source, third party inspection is impossible. At present the industry does not insist on third party inspection or certification, but it's possible that as open source models become more popular then expectations of audits will rise.

An often-quoted example of this in real life is the Interbase product from Borland (Inprise). This[CERT advisory notice](http://www.cert.org/advisories/CA-2001-01.html) carries the following summary:

"Interbase is an open source database package that had previously been distributed in a closed source fashion by Borland/Inprise. Both the open and closed source versions of the Interbase server contain a compiled-in back door account with a known password."

The back door account was discovered when Borland released the source code of the software for public use. One is left wondering how many other software projects contain similar features — and it's not possible to know if the source code remains closed.

[CONECTA2000](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#conectapaper) notes:

"We can easily see that open source software has a distinct advantage over proprietary systems, since it is possible to easily and quickly identify potential security problems and correct them. Volunteers have created mailing lists and auditing groups to check for security issues in several important networking programs and operating system kernels, and now the security of open source software can be considered equal or better than that of desktop operating systems. It has also already been shown that the traditional approach of security through obscurity leaves too many open holes. Even now that the Internet reaches just a part of the world, viruses and cracker attacks can pose a significant privacy and monetary threat. This threat is one of the causes of the adoption of open source software by many network-oriented software systems."

One example of third-party auditing is the [Linux Security Audit](http://lsap.org/) project which performs by-eye auditing of open source software packages and maintains a database of results.

At present company auditors are not known to be asking about the suitability and auditability of key software packages which support crucial business processes. A plausible reason is that they simply don't know that it **can** be done. That situation is not guaranteed to continue.

## Cost

Most current Open Source projects are also available free of royalties and fees, leading to the confusion around the commonly used term `free software'. Regrettably the English language does not have separate concepts for free-of-charge and free as in unconstrained; other languages are better equipped to describe the difference between `freedom' and `free of charge' (**libre** vs. **gratis**). Proponents of free software licences tend to emphasise liberty over cost although in practice the main open source projects are free in both senses of the word.

From a business perspective the purchase cost of software is only one factor; total cost of ownership (TCO) is what really matters. Other things being equal, the solution with lowest TCO is usually the most desirable one. Arguments in favour of low TCO for open source software include:

* Possibly zero purchase price
* Potentially no need to account for copies in use, reducing administrative overhead
* Claimed reduced need for regular upgrades (giving lower/nil upgrade fees, lower management costs)
* Claimed longer uptimes and reduced need for expensive systems administrators
* Near-zero vulnerability to viruses eliminating need for virus checking, data loss and downtime
* Claimed lower vulnerability to security breaches and hack attacks reducing systems administration load
* Claimed ability to prolong life of older hardware while retaining performance

Some longer-term claims are more difficult to substantiate yet they need to be taken into account:

* Better adherence to standards permits competition in the market, reducing vendor lock-in and consequent monopoly pricing
* Availability of source code provides greater continuity and security against
  + Financial collapse of vendors of key products
  + Vendors choosing to withdraw support for unprofitable products
* Protection against being required to fit your IT strategy to the cash needs of your software supplier

Unfortunately in this area there are numerous claims and counter claims. Reliable TCO information is practically unobtainable, although the case studies which form part of this guide provide a large amount of circumstantial evidence in favour of the argument. Most businesses will have to chose the argument on on its merits and choose to back the use of Open Source software where it seems most likely to provide either a clear cost win, or valuable leverage over entrenched suppliers.

## Flexibility and Freedom

In a business context, software flexibility is about being able to choose solutions suitable for the needs of the users. Many commercial software products will claim flexibility as a built-in feature and some will undoubtedly be correct. Our view is that that flexibility should really mean business flexibility, so that as requirements in the business change, solutions should not be unreasonably constrained by software. In particular, we view this as being especially important in the area of infrastructure components — the architecture of the IT solution rather than any one package.

To obtain flexibility at the architectural level, experience shows that it is often best to pick tried and trusted standards for interworking. If that is done, then best-of breed solutions can be selected for particular components within the architecture. Provided that the solutions can interwork suitably, the business should be able to avoid lock-in to a particular supplier and over-dependency. This is notoriously hard to manage, requiring a real act of will from management. What happens most often is that a vendor will make a `feature sale', emphasising something which cannot be done through the standard infrastructure. If they succeed then the business can become dependent on that particular solution and unable to choose alternatives at a later date. Any astute vendor will attempt to do this, only vigilant managers can avoid the lock-in that follows. Proprietary data formats are a particularly good tool for vendors to use. If they can establish a bridgehead, their competition will not only have to provide competing functionality, but also data conversion tools from a (typically) undocumented or even protected format.

Open Source projects have very little motivation to attempt this kind of lock-in strategy. Since there is no commercial benefit to be had, adherence to de-jure or de-facto standards (where they exist) is typically high. Where standards for interworking do not exist, the fact that the source code is published means that proprietary data formats can't be used to manipulate lock-in. This at least partly explains the relative success of Open Source software in infrastructure areas. Many vendors have tried to create web servers to compete with Apache, but because the network protocol used between browsers and the web server is well specified they have had to compete on quality or features rather than through more insidious tactics. Any vendor that controlled the lions' share of the browser **and** the server market would feel strongly tempted to exclude competitors by proprietary extensions to the HTTP protocol if they thought they could get away with it. No single vendor has yet managed to control both ends of this equation to a great enough degree.

"Open Source software tends to be free of dependency on related products. Purchasers often perceive that the product works best with other products from the same manufacturer. Open Source software offers its users greater freedom to purchase other products, avoiding lock-in to particular manufacturers." ([QINETIQ2001](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#qinetiq2001)).

Business users are most likely to obtain long-term flexibility through the careful choice of standards for interworking and data exchange, followed by vigilance to ensure that freedom from proprietary lock-in is maintained in critical areas. The drawback is that standards inevitably lag in terms of glamorous features, making the feature-based ploy an easy one for proprietary vendors to use. Open Source products are strong in this area, not only from the point of view of adherence to standards but also by helping to mitigate against insidious lock-in if they are chosen as core infrastructure components.

"It is the use of proprietary standards and protocols that effectively mandates the purchase of further products from the same supplier. Mandating the use of open internet standards (as in the e-GIF) rather than proprietary formats, and developing XML-based data definitions, for intra-Government, and Government-to-Citizen interoperability, is a practical approach to controlling the [proprietary lock-in]". ([QINETIQ2001](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#qinetiq2001)).

The choice is not easy. Many weaker managements simply roll-over and buy from a single supplier, but they pay a high price.

Open Source software provides further flexibility through freedom.

Freedom from a single vendor

Software vendors can go out of business, and they can arbitrarily decide to cease development of a product. How would your business cope if it relied on such a product? Open-source software allows you to retain not just the right to use the software you already have, but the ability to continue to use it as your needs change.

Freedom to modify your software

You aren't limited to what one company believes you need. Proprietary software vendors must cater for many different companies, predominantly their own. Open-source software can be tailored for the way **you** do business. It is usually within the resources of all but the smallest companies to modify Open Source software to suit their own needs (and potentially then to make those enhancements available as a public good). If in-house development skills don't exist, a short email to the project's mailing list will usually find a suitable consultant.

## Support and Accountability

It may appear counter-intuitive at first, especially to someone used only to dealing with proprietary software, but whilst the models for obtaining support and accountability for Open Source software are clearly different, the Open Source outcome is generally better than for all but unusual cases of user-vendor relationships.

One of the most common counter-arguments to the use of Open Source software is characterised as the ‘who do I sue?’ question; in other words, who is liable if the software doesn't work?

This argument seems plausible in theory. Unfortunately, though, that isn't what aplies in all but the rarest of circumstances. A moment's inspection of typical EULA (End User Licence Agreements) will dispel that myth. All usual software licences explicitly disclaim responsibility or liability for anything more serious than defects on the distribution medium, with the responsibilities being a one-way street and resting on the user, not the supplier. Proprietary software licences are intended to absolve the vendor of liability for almost any problem you may incur. Major vendors have large legal teams whose job it is to prevent the vendor from being liable for anything. This does not seem unreasonable – the vendor wants to sell software, not spend months or years at a time defending protracted tort and liability suits.

Open-source software does not differ from proprietary software in this respect. Open-source licences typically disclaim all liabilities and warranties, including such basic warranties as merchantability and fitness for purpose. Those in the know, who have adopted Open Source Software already, will shrug their shoulders and choose the practical benefits of increased reliablility and security over illusory options to sue or pursue other remedies from a negligent vendor.

Detractors of open-source software quite rightly point out that the free licence to use the software includes no support contract. But they neglect to mention the other side of that issue: many proprietary software licences have no support included either. Indeed, the majority of mass-market proprietary software support is aimed at hand-holding for inexperienced users. Just as proprietary vendors will sell support contracts with agreed service levels, suppliers and third parties will provide support for open-source software. An example of this is the [Gnat](http://www.gnat.com/)translator for the Ada programming language, as reported in [[CONECTA2000]](http://open-source.gbdirect.co.uk/migration/migration_guide_references.html#conectapaper) and elsewhere.

"ACT Europe was founded in 1996 to provide support for commercial, industrial and military uses of the GNAT Professional Ada 95 28 development environment in Europe. It was founded jointly by Ada Core Technologies Inc (ACT), and by the European members of the GNAT Ada 95 project. ACT is a privately held corporation and was founded in August 1994 by the principal authors of the GNAT Ada 95 compiler system. ACT was founded without any initial capital other than some small loans from the principles, and has existed entirely from revenue from its inception. ACT claims to be currently profitable. The people involved in both ACT and ACT Europe (from now on, ACT) have been working with Ada for over twenty years, starting with the development of a working Ada compiler for preliminary Ada in 1979, and the first validated Ada 83 system in 1983. This work was done at New York University by a team dedicated to the technical success of the Ada language, which moved to ACT after its foundation. GNAT is the most widely used Ada 95 development system, available on many platforms, from workstations and PCs to bare boards. Ada solutions using GNAT encompass conventional, real-time, embedded, and distributed systems applications. The GNAT technology has always been based on free software, and more specifically on the GNU toolset. The GNAT compiler is integrated with the GCC (the GNU C compiler) back-end. The GNAT debugger is based on GDB (the GNU debugger) that has been adapted for Ada 95. GLIDE, the GNAT integrated development environment, is based on Emacs, which has been adapted and complemented to create a friendlier and complete environment. The GTKAda GUI technology leverages on the GTK graphical toolkit and provides an advanced GUI builder. These are a few but significant examples of the technological offering of ACT. All the products being distributed by ACT are free software, usually under a GPL or LGPL-like licence. Since its foundation, ACT has established strategic partnerships with hardware and software manufacturers providing Ada 95 capabilities. In the former category is ACT's relationship with Silicon Graphics, Inc., whose new SGI Ada 95 product is based on GNAT. As of 1999, SGI sold over a billion dollars of Ada related software and equipment. ACT has relationships with several other hardware manufacturers such Compaq and Hewlett Packard. Commercial customers of ACT Europe include Aerospatiale, Alenia, BNP, Boeing, British Aerospace, Canal+, CASA, Dasa, Ericsson, Hughes, Lockheed, TRW, etc."

To add some context to that quote above, Ada is a language designed **specifically** for military, industrial and aerospace mission-critical and safety-critical systems where human life is routinely risked if software systems do not function correctly.

Some software vendors produce free software, and obtain large parts of their revenue from service and support (e.g. [Zope](http://www.zope.com/)). In other situations, open-source consultants will provide training and/or support for software they recommend. However, the fundamental advantage of open-source software when it comes to support is that it's always possible to retain a company to provide support. Because the source code is freely available, organisations are not limited to obtaining support from the authors. There is no restriction on other suppliers learning enough about the software to provide adequate support whenever demand exists.

Speaking from a personal perspective, the authors feel obliged to point out that when software is as reliable as Apache, MySQL and similar projects, support dwindles as an issue. When software works reliably, support for it ceases to be a frontline concern.

# **10 Reasons Open Source Is Good for Business**

[40COMMENTS](http://www.pcworld.com/article/209891/10_reasons_open_source_is_good_for_business.html#comments)

By [Katherine Noyes](http://www.pcworld.com/author/Katherine-Noyes/), PCWorld

* Nov 5, 2010 1:00 PM

With the many business and government organizations that now use open source software such as Linux, it's becoming increasingly clear that price is not the only advantage such software holds. If it were, companies that adopted it during the Great Recession would surely have switched back to the expensive proprietary stuff as soon as conditions began to ease, and that's clearly [not the case](http://www.pcworld.com/businesscenter/article/207479/linux_is_on_the_rise_for_business.html).

Rather, free and open source software (FOSS) holds numerous other compelling advantages for businesses, some of them even more valuable than the software's low price. Need a few examples? Let's start counting.

**1. Security**

It's hard to think of a better testament to the [superior security](http://www.pcworld.com/businesscenter/article/202452/why_linux_is_more_secure_than_windows.html) of open source software than the recent discovery by Coverity of a number of defects in the Android kernel. What's so encouraging about this discovery, as I [noted](http://www.pcworld.com/businesscenter/article/209419/88_highrisk_defects_found_in_android_kernel.html) the other day, is that the only reason it was possible is that the kernel code is open to public view.

Android may not be fully open source, but the example is still a perfect illustration of what's known as "Linus' Law," named for Linus Torvalds, the creator of Linux. According to that maxim, "Given enough eyeballs, all bugs are shallow." What that means is that the more people who can see and test a set of code, the more likely any flaws will be caught and fixed quickly. It's essentially the polar opposite of the "security through obscurity" argument used so often to justify the use of expensive proprietary products, in other words.

Does the absence of such flaw reports about the code of the iPhone or Windows mean that such products are more secure? Far from it--quite the opposite, you might even say.

All it means is that those products are closed from public view, so no one outside the companies that own them has the faintest clue how many bugs they contain. And there's no way the limited set of developers and testers within those companies can test their products as well as the worldwide community constantly scrutinizing FOSS can.

Bugs in open source software also tend to get fixed immediately, as in the case of the Linux [kernel exploit](http://www.pcworld.com/businesscenter/article/205867/linux_kernel_exploit_gives_hackers_a_back_door.html) uncovered not long ago.

In the proprietary world? Not so much. Microsoft, for example, typically takes weeks if not months to patch vulnerabilities such as the recently discovered [Internet Explorer zero-day flaw](http://www.pcworld.com/businesscenter/article/209661/what_you_need_to_know_about_new_ie_zeroday.html). Good luck to all the businesses using it in the meantime.

**2. Quality**

Which is more likely to be better: a software package created by a handful of developers, or a software package created by thousands of developers? Just as there are countless developers and users working to improve the security of open source software, so are there just as many innovating new features and enhancements to those products.

In general, open source software gets closest to what users want because those users can have a hand in making it so. It's not a matter of the vendor giving users what it thinks they want--users and developers make what they want, and they make it well. At least one recent study has shown, in fact, that technical superiority is typically the [primary reason](http://www.pcworld.com/businesscenter/article/207479/linux_is_on_the_rise_for_business.html) enterprises choose open source software.

**3. Customizability**

Along similar lines, business users can take a piece of open source software and tweak it to suit their needs. Since the code is open, it's simply a matter of modifying it to add the functionality they want. Don't try that with proprietary software!

**4. Freedom**

When businesses turn to open source software, they free themselves from the severe vendor lock-in that can afflict users of proprietary packages. Customers of such vendors are at the mercy of the vendor's vision, requirements, dictates, prices, priorities and timetable, and that limits what they can do with the products they're paying for.

With FOSS, on the other hand, users are in control to make their own decisions and to do what they want with the software. They also have a worldwide community of developers and users at their disposal for help with that.

**5. Flexibility**

When your business uses proprietary software such as [Microsoft Windows](http://www.pcworld.com/businesscenter/article/207196/why_windows_is_bad_for_business.html) and Office, you are on a treadmill that requires you to keep upgrading both software and hardware ad infinitum. Open source software, on the other hand, is typically much less resource-intensive, meaning that you can run it well even on older hardware. It's up to you--not some vendor--to decide when it's time to upgrade.

**6. Interoperability**

Open source software is much better at adhering to open standards than proprietary software is. If you value interoperability with other businesses, computers and users, and don't want to be limited by proprietary data formats, open source software is definitely the way to go.

**7. Auditability**

With closed source software, you have nothing but the vendor's claims telling you that they're keeping the software secure and adhering to standards, for example. It's basically a leap of faith. The visibility of the code behind open source software, however, means you can see for yourself and be confident.

**8. Support Options**

Open source software is generally free, and so is a world of [support](http://www.pcworld.com/businesscenter/article/207958/how_to_get_support_for_open_source_software.html) through the vibrant communities surrounding each piece of software. Most every Linux distribution, for instance, has an online community with excellent documentation, forums, mailing lists, forges, wikis, newsgroups and even live support chat.

For businesses that want extra assurance, there are now paid support options on most open source packages at prices that still fall far below what most proprietary vendors will charge. Providers of commercial support for open source software tend to be more responsive, too, since support is where their revenue is focused.

**9. Cost**

Between the purchase price of the software itself, the exorbitant cost of mandatory virus protection, support charges, ongoing upgrade expenses and the costs associated with being locked in, proprietary software takes more out of your business than you probably even realize. And for what? You can get [better quality](http://www.pcworld.com/businesscenter/article/202563/11_free_linux_apps_your_business_needs_now.html) at a fraction of the price.

**10. Try Before You Buy**

If you're considering using open source software, it will typically cost you nothing to [try it out](http://www.pcworld.com/businesscenter/article/205640/4_ways_to_give_desktop_linux_a_testdrive.html) first. This is partly due to the software's free price, and partly due to the existence of LiveCDs and Live USBs for many Linux [distributions](http://www.pcworld.com/businesscenter/article/204310/how_to_choose_a_desktop_linux_distribution.html), for example. No commitment required until you're sure.

None of this is to say, of course, that your business should necessarily use open source software for everything. But with all the many benefits it holds, you'd be remiss not to consider it seriously.

# **ADVANTAGES AND DISADVANTAGES OF OPEN SOURCE**

Posted August 28, 2012 By [CloudTweaks](http://cloudtweaks.com/author/cloud77/" \o "View all posts by CloudTweaks)

* **25**

# **Advantages And Disadvantages Of Open Source**

Open source is a fairly new concept that has gained huge popularity in the field of IT in recent years. This is mainly because [open-source software](http://www.cloudtweaks.com/2012/08/open-source-software-in-cloud-applications/) is free to use – its greatest advantage. As it is developed by a non-profit community, it has some disadvantages as well.

Open-source software is free to use, distribute, and modify. It has lower costs, and in most cases this is only a fraction of the cost of their proprietary counterparts.

Open-source software is more secured as the code is accessible to everyone. Anyone can fix bugs as they are found, and users do not have to wait for the next release. The fact that is continuously analyzed by a large community produces secure and stable code.

Open source is not dependent on the company or author that originally created it. Even if the company fails, the code continues to exist and be developed by its users. Also, it uses open standards accessible to everyone; thus, it does not have the problem of incompatible formats that exist in proprietary software.

Lastly, the companies using open-source software do not have to think about complex licensing models and do not need anti-piracy measures like product activation or serial number.

## ****Disadvantages****

The main disadvantage of open-source software is not being straightforward to use. Open-source operating systems like Linux cannot be learned in a day. They require effort and possibly training from your side before you are able to master them. You may need to hire a trained person to make things easier, but this will incur additional costs.

There is a shortage of [applications](http://research.cloudtweaks.com/cloud-vendors/) that run both on open source and proprietary software; therefore, switching to an open-source platform involves a compatibility analysis of all the other software used that run on proprietary platforms. In addition, there are many ongoing parallel developments on open source software. This creates confusion on what functionalities are present in which versions.

Lastly, many of the latest hardware are incompatible to the open-source platform; so you have to rely on [third-party drivers](http://www.cloudtweaks.com/2012/08/what-does-the-wizard-of-woz-have-against-cloud-computing/).

[](http://cloudtweaks.media/)

## ****Advantages****

The decision of adopting open-source software should not be taken just on the basis of the low-cost involved. It entails a detailed analysis and understanding of the requirements before switching to open source to get full benefits of it.

[en source software (OSS)](http://tech.co/tag/open-source-software) are accessible under a software authorization that enables individuals to access the source code and customize it according to their needs so providing the capability to tailor the software for different jobs. The program license keeps the right of the individual to modify and customize it in any way they desire. This feature is the major difference among open source software and blocked source software model. The blocked source model enables only the developer of the application to access and customize the program according to their requirements.

A few of the advantages of open source program are:

## 1. Customization

The capability to modify the program according to the needs is the major difference among closed source and open source software. The choice of modifying the software enables developers to make a solution that particularly targets the needs of their customers. Enterprises and companies can extract the utmost advantages from this feature because they can get modified solutions for controlling their daily activities.

## 2. Freely Obtainable

One of the other excellent advantages of making use of this software is that it is generally obtainable for free. Users simply need to download the program from online and begin using it. This enables the user to use the saved resources for program development and other activities. This also assists in decreasing the overall expenses of the project.

## 3. Support

Generally, such popular programs are supported by a big community of program developers who are always accessible on the internet and assist out developers who face difficulties while making solution. These software generally come together with a well planned documentation that is extremely useful while setting up and operating the software.

## 4. Accessibility of Resources

A successful program is always in demand so nearly all software development organizations provide skilled resources that can exploit these software for making a meaningful and helpful solution for customer. The developers as well need to keep themselves updated and informed with up to date and famous technologies if they want to go on in the hard job environment.

## 5. Enterprise Administration Solutions

Each and every industry is different. The requirements and problems within each are dissimilar as much as the solutions needed for them. It is nearly impossible to create software that will wholly fulfill the needs of any one kind of business. WhiteSource open source management software can be modified according to the needs of the user can be modified according to business particular needs to make a solution that can meet the needs of specific organization.

At present there are a number of OSS accessible on the internet that can be openly downloaded. Some of the popular open source web technologies are [Java](https://java.com/download), PHP, [Ruby on Rails](http://rubyonrails.org/), Magento, WordPress, Joomla etc. these technologies together with a lot of other OS technologies enable the makers all over the world to make customer oriented solutions that are reliable and flexible. The OS technology is an advantage for web developing companies who are generally working on inflexible budgets and want cost effective-mediums for developing and offering quick and helpful solutions to their customers. As open source program projects tend to attract a lot of developers and the most excellent code contributions are usually accepted, the feature of OSS tends to be better than proprietary software solutions which usually only have handful of makers contributing to the code base. As the source code is freely obtainable, it is free to everybody to check for safety vulnerabilities.

# Advantages of open source software

**Page Content**

[Lesser hardware costs](https://www.outsource2india.com/software/articles/open-source-software.asp#heading1)[High-quality software](https://www.outsource2india.com/software/articles/open-source-software.asp#heading2)[No vendor lock-in](https://www.outsource2india.com/software/articles/open-source-software.asp#heading3)[Integrated management](https://www.outsource2india.com/software/articles/open-source-software.asp#heading4)[Simple license management](https://www.outsource2india.com/software/articles/open-source-software.asp#heading5)[Lower software costs](https://www.outsource2india.com/software/articles/open-source-software.asp#heading6)[Abundant support](https://www.outsource2india.com/software/articles/open-source-software.asp#heading7)[Scaling and consolidating](https://www.outsource2india.com/software/articles/open-source-software.asp#heading8)

Today open source software has become critical for almost every organization. Almost everything requires open source software, be it telecommunication systems, inventory, accounting, personal productivity applications, contact management and operating systems amongst others. At Outsource2india, we have experienced and skilled software engineers who can proficiently build a software system by using open source software. With our expertise in java development, we can also develop application blocks. We also use our system integration services to make sure that the new application that we create can be easily integrated with your existing systems. **Outsource open source software development to O2I and benefit from high-quality services at a cost-effective price**.

Open source software can have a major impact on your entire organization. There are several advantages of using open source software. The following are a list of the advantages of opting for open source software.

## 1. Lesser hardware costs

Since Linux and open source solutions are easily portable and compressed, it takes lesser hardware power to carry out the same tasks when compared to the hardware power it takes on servers, such as, Solaris, Windows or workstations. With this less hardware power advantage, you can even use cheaper or older hardware and still get the desired results.

## 2. High-quality software

Open source software is mostly high-quality software. When you use the open source software, the source code is available. Most open source software are well-designed. Open source software can also be efficiently used in coding. These reasons make open source software an ideal choice for organizations.

## 3. No vendor lock-in

IT managers in organizations face constant frustration when dealing with vendor lock-ins'. Lack of portability, expensive license fees and inability to customize software are some of the other disadvantages. Using open source software gives you more freedom and you can effectively address all these disadvantages.

## 4. Integrated management

By using open source software, you can benefit from integrated management. Open source software uses technologies, such as, common information model (CIM) and web based enterprise management (WBEM). These high-end technologies enable you to integrate and combine server, application, service and workstation management. This integration would result in efficient administration.

## 5. Simple license management

When you use open source software, you would no longer need to worry about licenses. Open source software enables you to install it several times and also use it from any location. You will be free from monitoring, tracking or counting license compliance.

## 6. Lower software costs

Using open source software can help you minimize your expenses. You can save on licensing fees and maintenance fees. The only expenses that you would encounter would be expenditure for documentation, media and support.

## 7. Abundant support

You will get ample support when you use open source software. Open source support is mostly freely available and can be easily accessed through online communities. There are also many software companies that provide free online help and also varied levels of paid support. Most organization who create open source software solutions also provide maintenance and support.

## 8. Scaling and consolidating

Linux and open source software can be easily scaled. With varied options for clustering, load balancing and open source applications, such as email and database, you can enable your organization to either scale up and achieve higher growth or consolidate and achieve more with less.

## Outsourcing open source software development services to O2I

At O2I, we can provide you with the perfect open source software solution that effectively meets the requirements of your organization. We can help you assess, design and develop an open source solution that is customized for your organization. Outsource to O2I and get access to professional, cost-effective and high-quality services.

Outsource open source software development to O2I and give your organization a competitive edge!

# **Benefits of Open Source Software**

The Open Source model harnesses the power of distributed peer review and transparency to create high-quality, secure and easily integrated software at an accelerated pace and lower cost.

Primary benefits of open source software are:

* Choice: By its very nature, the source code of open source software is available to all, meaning that no one company owns the software. Any company can build upon open source software. It may be operated and maintained by multiple vendors, reducing both barriers to entry and exit. A customer can easily choose another vendor because they are not locked in to one vendor’s offering.
* Reliability: Open source is peer reviewed software, which leads to more reliability. The infrastructure of the Internet is largely composed of open-source programs such as DNS, sendmail, Apache and languages such as HTML and Perl. They have proven to be both reliable and robust under the most strenuous conditions, namely the fast growth of the Internet.Low-cost: Open source software is often developed through community forums and collaboratives. Developers volunteer their time and expertise, and are coordinated by fewer paid programmers. The lower overhead costs translates into substantial savings, as does the fact that open source software typically does not have a per-seat licensing cost. According to Gartner, open source is even more attractive to businesses during tough economic times. In its 2009 enterprise software spending forecast, the industry analyst predicts that enterprises seeking to cut costs will be drawn to open source software, virtualization technologies and, because of tighter travel budges, unified messaging and collaboration technologies.
* Security: Open source enables anyone to examine software for security flaws. The continuous and broad peer-review enabled by publicly available source code improves security through the identification and elimination of defects that might otherwise be missed. Gartner for example, recommends the open source Apache Web server as a more secure alternative to closed source Internet Information servers. The availability of source code also facilitates in-depth security reviews and audits by government customers.
* Fast deployment: With open source software consumers needn’t wait years to deploy a solution. Open source software can be “test driven” prior to procurement, and is particularly suitable for inter-agency collaboration, rapid prototyping and experimentation. Both known and unanticipated users can be rapidly provisioned.

# What Is Free Open-Source Software – Pros & Cons, Best Options

http://1.gravatar.com/avatar/7ff1439f266170c2132bf81ddc56e7a8?s=40&d=mm&r=pgBy David Gomez   
Posted in: [Technology](http://www.moneycrashers.com/category/lifestyle/technology/)

[**Comments**0](http://www.moneycrashers.com/free-open-source-software-definition/#respond)

**Best Banking Deal**: Sign up for a [Chase Total Checking® account](http://www.moneycrashers.com/rec/chasebank-totalchecking-beforetext/) and a [Chase SavingsSM account](http://www.moneycrashers.com/rec/chasebank-savings-beforetext/) today and get up to a**$250 bonus**. Get the details and your bonus coupon [here](http://www.moneycrashers.com/rec/chasebank-checkingsavings-beforetext/).

Tools are an important part of any job. But what if the tools you need are out of your price range?

When searching for industry-leading software, people often end up with their jaws on the floor due to sticker shock. Software programs that large organizations use may be ideal, but many small businesses and self-employed individuals cannot justify spending thousands of dollars to use a package of programs, no matter how useful.

Fortunately, in many instances you can find software tools that meet your specific business needs – all for free.

## Free Software

The free software movement has been in existence since 1985, and according to the [Free Software Foundation](http://www.fsf.org/), it can be summed up thusly:

“Free software is a matter of liberty, not price. Think of ‘free’ as in ‘free speech,’ not as in ‘free beer.’ Free software is a matter of the users’ freedom to run, copy, distribute, study, change, and improve the software.”

This means that free software gives users the ability to have total access to a program’s source code, whereas conversely, proprietary software does not allow users access to a program’s source code, nor to make changes to it.

### Advantages

Often, free software has the same features as its high-priced proprietary counterparts – and that is only one of the many advantages:

* **Free to Download**. Would you rather pay upwards of a thousand dollars for the program you need, or nothing?
* **Many Choices**. There are more than 6,500 pieces of free software available today.
* **Large Communities**. Many free software programs have large and active communities online that offer support to users of free software via blogs and forums.
* **Supportive of Social Movement**. The use of free software supports a movement that believes computer users should be free from technology that conducts surveillance and gathers information about what users do on their devices.
* **Shared Improvements**. Users are allowed to have source code, study it, make functional changes to a program, and redistribute the modified software to others in any way they choose. A crafty entrepreneur can take advantage of this to save money by developing his or her own custom business software.

### Disadvantages

Despite the upsides, there are also disadvantages to free software:

* **No Guaranteed Support**. Some free software programs don’t have a large user base, and therefore the user support for certain programs can be lacking or nonexistent.
* **Inconsistent Updates**. Since many members of the free software community develop the code in their spare time as unpaid volunteers, there is a chance that some of the programs in the free software directory haven’t been updated in a while, and may not function properly on newer operating systems.
* **Varying Interfaces**. Some free software programs have a much different user interface than their commercial counterparts, and can have a steep learning curve.

Open source software has made waves in the tech world creating a movement for software to be more transparent and accessible to all. From this movement, strong advocates for and against open source software have sprung up creating a heated debate between these two groups.

We have gathered four articles that take a look at both sides of the open source software debate. Leave your opinion in the comments below.

## 1. [The Open Source Conundrum: 3 Benefits and Drawbacks](https://www.americanexpress.com/us/small-business/openforum/articles/the-open-source-conundrum-3-benefits-and-drawbacks-to-consider/)

Historically, open source software was a labor of love for hobbyists and hackers. Today, open source is a multi-billion dollar industry and many companies opt to use open-source software as their primary software platforms— from word processing to operating systems. Granted, a good chunk of these businesses are not-for-profit companies but for-profit businesses are readily adopting open source in increasing numbers. Should you?

The article looks at 3 benefits and drawbacks of open source software.

**Benefits**:

1. The license price
2. The stability
3. The competitive edge

**Drawbacks**:

1. It’s not the best answer in most cases
2. Legal implications
3. Security issues

## 2. [Why Open Source Sucks](http://www.hackerfactor.com/blog/index.php?/archives/415-Open-Source-Sucks.html)

Dr. Neal Krawetz has a bone to pick with open source software tools. The blog post is his personal rant on why open source sucks, but it offers an honest insight into the drawback of open source software.

Here are a few points he makes in his article Why Open Source Sucks:

#### **1. Prima Donnas**

Krawetz points out that in his career as a technical engineer in a Fortune-500 company, there was always a buffer between his blunt comments and the customer. In open source software, you the customer get to meet the blunt, rude engineers. You will meet the ones on a power trip, others are trolls and a handful of friendly, helpful ones.

#### **2. Real Costs**

Open source is really good if you’re cheap (i.e., don’t want to spend one penny). However, just being free is not the same as “no cost”. Since open source code generally doesn’t have the same budget as commercial software, there is usually less effort given to usability, documentation, and even development. You can expect to burn time fighting with the installations and dealing with the learning curve.

## 3. [Why Open Source Rocks](http://www.hackerfactor.com/blog/index.php?/archives/416-Open-Source-Rocks.html)

On the flip side, Krawetz gives the argument that open source is pretty good. It is easy enough to criticize open source software about things that require a good amount of marketing, documentation, support, and legal staff. Most open source projects have little or no funding and are operated by a small handful of programmers.

“As far as I know, there isn’t a free, open source usability lab (unless you count the “end user”). In contrast, commercial software usually has the budget for market research, usability testing, and even a couple of support personnel. But with size and funding come other limitations, and it is here that open source software really shines.” says Krawetz.

Here’s a few points he makes to support open source:

#### **1. Transparency**

“Open source” does not mean “free”. There are plenty of open source projects that are shareware, beerware, and postcardware. “Open source” only means that the source code is available. By being available, the functionality becomes transparent; anyone can see what it does and how it works.

#### **2. Sparks Innovation**

Competition comes from many areas. However, most commercial projects that I have been on were marketing driven. The people in marketing have a checklist of features and the developers must implement the functionality. As a result, products compete with each other. If Product A has a new feature, then Product B must incorporate something similar.

With open source code, you don’t have market-driven functionality. If a function is cool, then it gets implemented. If it sucks, it gets left out. As a result, you can get some really novel functionality from open source.

## 4. [The True Cost of Free Testing Tools](http://www.neotys.com/blog/the-true-cost-of-free-testing-tools/)

Open source testing tools carry a certain appeal to testers – there’s no doubt about that. However, there is a pretty intense debate out there between open source testing tool advocates and in-house/[vendor testing tool](http://www.neotys.com/neoload/overview) advocates. Both camps provide equally valid points.

But, we have to wonder, what is the true cost of using a free open source testing tool? Here are some points we came up with for why free open source software tools aren’t really free.

#### **1. Free tools might not be as free as you think**

Free testing tools are like free puppies. Think about it. You get a puppy as a gift, but that puppy lives for 18 years or so and requires food, vet visits, training classes, visits to the groomers and a whole lot more basic necessities to keep your furry friend healthy and happy.

#### **2. Lack of professional tech support**

There are open source communities out there that are very responsive and many argue new features are rolled out faster than vendors. But, you can’t really count on the community too heavily because it’s not their job. They aren’t getting paid to work on the tool, and if you want the tool to have a feature that does X-Y-Z, you have to wait until someone in the community decides to implement that feature

### **List of Advantages of Open Source Software**

**1. Cheaper than commercially marketed products.**  
According to studies, open source software collectively help business owners save around $60 billion a year. This might seem unbelievable at first, but it’s not really surprising since these programs are developed to be accessible to anyone (especially those who can’t afford to buy commercial products). For one thing, they’re usually offered for free and don’t require you to pay for any additional copy you download. Since many of these programs are created to work with almost any type of computer, they can also help you lengthen the life of your old hardware and avoid the need to replace them every now and then.

**2. Created by skillful and talented people.**  
Large and well-established software companies have the financial capability to hire the best talent in the business to create their products. Because of this, many people opt to buy computer programs from these firms because they think they’ll get great value for their money by doing so.

The fact that big companies hire experienced, fully trained and highly qualified people is true. But what consumers have to know is this: not all software developers care deeply about money. Sure, most of them do get a job to have a steady income and be able to financially support themselves and their family. However, a lot of these experts don’t just base their worth on the salary they earn; rather, they strive to build a program that will earn the admiration of their peers and hopefully make a difference in the world.

This is why many software developers turn to open source products as an outlet for their ideas and creativity. By doing so, they won’t be confined by the rigid rules of the corporate world, and they’ll have the freedom to experiment and come up with high-quality programs. These, in turn, benefit consumers since they’ll have access to world-class and state-of-the-art software without having to pay too much.

**3. Highly reliable.**  
There are two main reasons why open source software are reliable. First of all, they’re developed chiefly by skillful and talented experts who do their best to create high-quality programs. Second, they’re worked on by tens or hundreds of people, which means there are numerous eyes that can monitor for the presence of bugs and many pairs of hands that can fix these defects within the shortest amount of time. Both of these factors lead to products that have excellent quality and helpful features and perform well most (if not all) of the time.

**4. Help you become more flexible.**  
Since you’re not tied to a proprietary product, you don’t need to abide by a specific IT architecture that might require you to upgrade your software and even hardware often. Rather, you can mix and match your software and create a unique IT infrastructure that best suits your needs. There’s no need to fret since there’s a wide range of options in the market, so you only have to browse through them and pick one those that match your requirements and specifications. Can’t find anything you like? You can modify existing open sources software or hire someone who can do it for you.

Open Source software, like its name suggests, provides users with an open code that can be freely used, modified, and shared by everyone. You can take advantage of it either for your business, for a personal project or for educational purposes without paying a dime. Open source licenses can grant you the right to copy and redistribute the software to everyone you want. There are no set boundaries or any limitations.

### **How does it work?**

The source code is visible for all users and it can be modified for your own needs. This can be done if you have a proper knowledge and experience in programming languages. One of the ideas behind Open Source software is that users can be co-developers, suggesting how to improve it and helping to find out bugs. When someone detects a bug it reports it to the community and they release a patch as soon as possible.In most cases this release is faster than the Proprietary Software’s patches.

### **What are the advantages of Open Source Software?**

The main advantage of Open Source is that it is generally free and you do not have to pay for using it. This saves a lot of money for big corporations and even for ordinary users. Most licenses of the proprietary software systems are expensive and should be renewed every year. As we know it is really important for any big company to reduce the expenses as much as possible. When Corporations are happy with the used software they donate money to the communities in order to be able to continue their work and developing new versions.

Most open source applications have their own communities which are constantly evolving the software thus improving its quality and security. If a user finds a bug , he reports it to the community and they release a patch as soon as possible.

Open Source software can be adapt for your own business demands which can not be done with proprietary systems.In addition, it can also be modified for using with various systems. It does not obligate you to run it on a determined platform like some proprietary software. I personally love to choose my own direction and not to be forced by anyone. So far it all sounds like a clear choice, right? Well…not entirely…

### **Any disadvantages?**

One disadvantage of Open Source software is that most interfaces are not so user-friendly and easy to use. More attention tends to be paid to the functionality than the user interface, making some of the scripts a real pain for the novice. If you plan to incorporate such software in your company, keep in mind that your employees may need to be especially trained in order to operate it properly.

It is also very difficult to find drivers for some devices. This can turn into a problem because when you buy a new hardware with many functionalities, if drivers are not available for the current Operating System you cannot take advantage of these functionalities.

Another disadvantage of Open Source is the support service. In most cases you should rely on the corresponding community or pay for external supporting service, which is quite ok if you are using a popular software with lots of followers, but can be a heavy burden for less caught on applications. The visibility and easy access to the source code, which I mentioned earlier, can also be a double-edged sword as bad-intentioned people can find vulnerabilities and leave backdoors for future exploits. Some cross-site-scripting attacks are possible when the attacker finds a weak spot in your source code and they can even leave the malicious code dormant for a long time before executing it. A way to prevent that is regular updates and only download software, themes and plugns form reliable sources.

As a conclusion I would like to say that Open Source gives you a boundless freedom, but one that should be taken seriously. You can test, explore, tweak, modify almost anything, but if you neglect security you are just inviting the bad guys in your home…er…website. What is you experience with Open Source software? Or do you find proprietary ones better for your purposes? You can share your stories in the comments below.