

Can the Public Sector Benefit from Open source Principles?

Dr Colin Turner
Senior Lecturer, School of Engineering,
University of Ulster
c.turner@ulster.ac.uk

20th October 2008
NICS ICT Conference

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

- 1 Introduction
- 2 Consumption of FOSS
- 3 Creation of FOSS
- 4 Consumption of FOSS ideas

Context of this talk

This talk is mainly focused on exploring the relationships between **FOSS**, Free and Open Source Software, the broader issue of Open Standards and the public sector.

On the 15th October, Sir Reg Empey opened the Open Source Solution Centre in the SRC in Newry. His speech provided some useful local context.

Sir Reg Empey

Open Source Solution Centre, 15th October 2008

Introduction

Consumption of FOSS

Creation of FOSS

Consumption of FOSS ideas

Summary

“... Today’s event is a tangible illustration of further progress and is a timely development, given the strategic importance of Open Source software to the long term sustainability of our software sector.

It is clear that Free and Open Source Software will become fundamental to building and maintaining future market share. The FOSS model can therefore provide start-ups with a fast and efficient way to build a client base and thus gain market advantage. ”

Sir Reg Empey

Open Source Solution Centre, 15th October 2008

“... Open source also adds value to the Northern Ireland software industry by providing impetus to their product and process innovation, resulting in faster delivery.”

“... The days of conventional commercial software companies holding a monopoly position over society are over.”

“... I would like to reinforce the message that Open Source solutions will in the future represent an important contribution to building and maintaining market share for the business and other sectors. The fact that all of us are gathered here today demonstrates the commitment of government to embrace this new climate of opportunity.”

Neelie Kroes

European Union Competition Commissioner, 10th
June 2008

"I know a smart business decision when I see one - choosing Open Standards is a very smart business decision indeed."

"There is a democracy issue as well. No citizen or company should be forced or encouraged to use a particular company's technology to access government information. No citizen or company should be forced or encouraged to choose a closed technology over an open one, through a government having made that choice first."

Terms and their usage

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

It's worth noting at this early stage that we should not take it for granted that people understand the meaning of **free** or **open source** software. Even in the FOSS community, there are some divergences of opinion and usage here.

So, with apologies to those for whom this is all well understood, we shall introduce the definitions in most common usage.

Terms and their usage

The Free Software Foundation (<http://www.fsf.org>) defines free software as providing **you** with the following freedoms with software:

Freedoms in Free Software

- 0 the right to run it for any purpose;
- 1 the right to study it and adapt to your needs;
(which implies **open source**)
- 2 to redistribute it, so you can help others;
- 3 to release improvements, so everyone benefits.

Specifically, note, **money** is not mentioned, but freedom 2 and 3 normally imply freedom from cost.

Terms and their usage

Other types of software include

- **Open Source:**

Most people use this term synonymously with free, but some mischievous people use this to literally mean: read access to the source.

- **Custom :**

Software created “in house” which isn’t being licensed for distribution - closed source, but without ethical entanglements normally.

... and

Terms and their usage

- **Proprietary** :

The “normal” model, software that is licensed, often in extraordinary ways that grants very little in return for the money. Having said that, again money might not be involved at all here, a lot of “freeware” and “shareware” software still grant none of the freedoms mentioned above.

“Crowley had been extremely impressed with the warranties offered by the computer industry, and had in fact sent a bundle Below to the department that drew up the Immortal Soul agreements, with a yellow memo form attached just saying: ‘Learn, guys’”
Pratchett & Gaiman, Good Omens

Back to the main thread

There are a number of ways in which the public sector could, and perhaps should, relate to FOSS.

Relationships of the public sector to FOSS

- 1 As a **consumer of software**.
Clearly, the sector procures vast quantities of software, should some of it be FOSS?
- 2 As a **creator of software**.
Custom software is created in large quantities, we should examine the pros and cons of making it FOSS.
- 3 As a **consumer of ideas**.
The methodologies used in FOSS are starting to be used in industry. They promote team work and transparency.

Why not use more FOSS?

It seems obvious that the public sector should be in favour of FOSS, the software is usually free of charge, and you have a lot of rights over it to customize it to need, and as the software is normally based in open standards, vendor lock in is avoided.

The OGC guidelines **require** open source solutions to be considered on a value for money basis alongside proprietary ones.

Despite this, uptake of FOSS by the sector is very limited, and there are common reasons given for this. So it's useful to explore and consider these.

Why not use more FOSS?

Here are some viewpoints that are typical in the sector, some are more valid than others.

Perceived problems with uptake of FOSS

- **Quality** “FOSS is flakey”;
- **Support** “We need support contracts”;
- **Interoperability** “We need to use what the real world uses”;
- **Gaps** “There is no FOSS program to do X”;
- **Special Licenses** “But we get all this at a huge discount anyway”;
- **Budgets** “If we spend less, we become less important”;
- **Transparency and Accountability** “Pardon?”.

Why not use more FOSS?

The Quality Myth

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

FOSS products are seen as lacking in QA measures, flaky and unreliable. Of course, this is as true in the proprietary sphere, but there you simply can't do anything if the company goes out of business, loses interest etc..

This should be looked at on a case by case basis. Much of the corporate, educational and public sector infrastructure of the internet has historically run with large elements of rock solid reliable FOSS. For example the **Apache** web server.

Many **GNU/Linux distributions** as well as other major FOSS projects like **Mozilla (Firefox)** have elaborate QA and excellent stability.

Why not use more FOSS?

The Support Myth

Perhaps the greatest fallacy. Support networks for FOSS products are often extremely rich, and superior to those of proprietary products. For example compare Debian's "for free" support, with similar features available for MS Windows.

Also support **can** be paid for, from any number of different sources, unlike in the proprietary model where support is often only available from the vendor.

In other words, there is a free market to provide support and maintenance and you are quite free to fix issues in house if you choose.

There is also a belief you can sue proprietary vendors more easily in case of problems. Check the fine print, you usually cannot.

Why not use more FOSS?

The Interoperability Myth

It is often argued that compatibility with other software is a major issue. There are exceptions, but generally proprietary software is in fact the natural enemy of open standards, with vendors needing to have distinctiveness in formats and other things built into the business model. Therefore, they tend to produce interoperability within a sphere of products from the same company. This does not provide true choice.

By contrast, “open source” implies open standards, since anyone can examine the code to find out how things are done and stored. This makes it straightforward to build interoperable products.

Many FOSS solutions are based around the concept of modules that do a given job well, or in other words, there are a number of interoperable choices for each piece of functionality you want.

Why not use more FOSS?

Gaps

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

There are a few areas, but not many, where FOSS solutions don't exist. It's not the purpose of this talk to be unrealistic about that, but to suggest that all other things being equal the FOSS solution is the obvious choice.

Nevertheless, where the solution is in some cases to write the custom software to do the job, we should consider making the product FOSS.

Why not use more FOSS?

Budgets

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

It's an uncomfortable fact to face, but in large organisations with departmental divisions, it's not always in the interests of a given department to save money on procurement. This will be a hidden motive and is therefore hard to counter. It's important to appropriately reward savings, and allow for transfers of how money is spent so that other resources can be obtained instead.

Perversely, it is often claimed that FOSS does not produce cost savings, since it requires training costs. That's true of all software, but experience shows there is no greater need for FOSS training than any other.

Why not use more FOSS?

Transparency and Accountability

Introduction

Consumption
of FOSS

Creation of
FOSS

Consumption
of FOSS ideas

Summary

This seems to go over the heads of many, but a basic issue is having software that implements closed standards with public data. Who owns the data? How will you access it in 20 years time? What is the software actually doing with it?

When you commission software do you own any rights? The copyright to the code? If not, does the license mean only the original authors can change it for you?

These are concerns many simply don't have, **but they should!**

Creating FOSS

So why should we create FOSS in the public sector?

A better place to start would be why not? Frequently we need to implement what would normally be custom software, a special in-house solution to a problem. Are there any pros and cons to making this free software?

Creating FOSS

Problems in making custom software FOSS

- **Security**; a common perception is that making software open damages its security. This approach, **security through obscurity** is generally devalued nowadays in any case;
- **Extra Effort**; a project started in an open fashion from its outset should require no extra effort to have as an open source venture.

... continued

Creating FOSS

Benefits to making custom software FOSS

- **Gift Culture**; giving software away and making it publicly available normally acts as a powerful incentive to improve quality;
- **Open in a box**; even the impact of visibility of a project within its parent organisation cannot be overlooked. Other developers can provide good ideas, or learn them. A “joined up” philosophy is more likely.
- **Prestige**; if you are going to develop the software, and not exploit it with a proprietary license, then why not show off your good practice;

... continued

Creating FOSS

Benefits to making custom software FOSS

- **Dissemination**; thousands of lines of code can be more valuable than journal articles;
- **Low risk business model**; support can be provided on a consultancy basis, with no obligation.
- **Gain expertise from others**; encourage other organisations with the same problems to improve your software at little or no man power costs to you;
- **Distribute expertise**; if your core team leave your control, you still have expertise in the code existing elsewhere (sustainability).

In short, FOSS, is an excellent exploitation model.

Creating FOSS

But perhaps the greatest benefit to an organisation for creating FOSS is in embracing the philosophy itself.

Indeed, the philosophy can be extended far beyond software.

Can we learn from the FOSS philosophy?

Eric S. Raymond characterised FOSS development as divided into two methodologies.

The Cathedral, where development took place behind closed doors, and the finished product was released to the general world.

The Bazaar, where the whole development takes place in the public view, and was open to the community from the outset.

The Bazaar method is considered superior and has become the de-facto standard within FOSS. It is, for example, the method employed by the Linux kernel.

Can we learn from the FOSS philosophy?

So what's good about it?

The Bazaar method has some important strengths.

- development is completely transparent to all stakeholders;
- there is therefore a higher level of accountability;
- it is harder for problems to be hidden and emerge later;
- ideas can permeate in and out of the project from a wider community.

Richard Branson, in January this year, announced that the Virgin Galactic technical development would follow the open source philosophy.

FOSS activity in Northern Ireland and Beyond

- **open-ni network**, to represent all open source interests in Northern Ireland.
<http://www.open-ni.org>.
- **foss@ulster**, to host development at the University of Ulster.
<http://foss.ulster.ac.uk>.
- **Open Source Solution Centre at the SRC**
<http://www.src.ac.uk>.
- **Open Source Academy Case Studies**
<http://www.opensourceacademy.gov.uk/solutions/casestudies>

Summary

- 1 It's a **first rate** solution, not **second rate**
 - Google uses it and produces it extensively, so does Sun, HP and many others.
 - SMEs use it to allow agile startups.
- 2 Consumption of FOSS
 - OGC guidelines require FOSS to be considered.
 - It provides open standards and avoids vendor lock-in.
 - FOSS procurement in the public sector is limited by many misconceptions.
 - Most of these are easily debunked.
- 3 Creation of FOSS
 - The public sector “may as well” create FOSS. Why not?
 - There are plenty of benefits in doing so.
- 4 Borrowing FOSS methodologies
 - Increasingly seen as an agile, cheap, accountable and sustainable way to conduct business.