Linux and Open Source in Government Conference

Overcoming the Challenges Facing Open Source in Government

Con Zymaris
<conz@cybersource.com.au>

Cybersource

What we will cover

- Vectors into Government
- Open Source advantages for Government
- Open Standards: Why they are crucial
- Gaining efficiencies through code reuse
- Warranty and indemnification challenges for code *used* by Government
- Warranty and indemnification challenges for code *created* by Government

Vectors into Government

- There are three main ways in which Open Source can be deployed into government.
  - Switched-on government IT staffers download and deploy such software.
  - An agency issues a Request for Tender (RfT) through which such an Open Source solution is proffered by a responding vendor.
  - major government outsourcer introduces an Open Source solution as part of a spectrum of technologies that they deem appropriate.

Open Source Advantages for Governments

- Huge potential for code reuse - saving IT budgets.
- Open standards, open formats and open protocols - increasing vendor competition.
- Consistent with the theme of government being for the public good. Open Source is much like the concept of an IP commons.
- Reduces code decay and code orphaning - a long term risk reduction for government.

Challenge of Code Reuse

- According to many long-time government software developers and contractors, there are many wasted opportunities to re-use perfectly viable code.
- This is due to a lack of a code-sharing culture and the absence of an accepted process and legal framework by which to achieve code reuse.
- An Open Source development process is a solution which can address this issue.
- If one government agency is coding a system, making the final product of such efforts available to other government departments and agencies has a potentially great benefit multiplier.

Solution – Code Banks

- A code bank can be considered a managed repository of source code and resources, available to all eligible government IT staffers and related developers.
- Code *metadata* and documentation will provide the necessary elements to allow government developers to determine if extant code can be of use in new projects
- Legal issues can be resolved by making this code available only within specific constituencies, i.e to other parts of the Crown.
- The machinery for this type of code bank exists and has been shown to scale to over 150,000 developers and 50,000 software projects, namely SourceForge/Savannah.
- Code banks can be created, along with a culture of re-use and an increased emphasis on designing for such widespread code re-use, from the beginnings of a project. This is better software engineering.

The Price of Not Following Standards

- In 1895 Mark Twain was on a train trip from Sydney to Melbourne.
- Twain was gob-smacked that a few hours after midnight, in Albury, the passengers were woken and asked to switch to another train which would continue them on their journey. Non-standard train gauges were the reason. His comment:
  - "Think of the paralysis of intellect that gave that idea birth, imagine the boulder it emerged from, on some petrified legislator's shoulders."
- There were 22 different rail gauges in use around the country. This monstrous problem was only resolved last week, over 100 years later. Imagine the economic cost!

What is an Open Standard?

- Many long, complicated definitions. In pure business terms, however, it boils down to the following simple question set:
  - Can you get some system, in essentially functionally identical and inter-operable form, from numerous, competitive suppliers?
  - Can your staff migrate knowledge from one vendor's products to another?
  - Can you ensure no loss of your efforts (infrastructure, data, programs) in any such migration?
- If yes to all these these questions, then most likely there is an open standard in place for this procurement line.

An example of Open Standards

- Automobiles! Why?
  - You can train you staff to use an automobile from any one manufacturer and they will then also be able to use any other from any other.
  - Infrastructure you've built to work with one car (roads, traffic signals, mechanics, petrol stations, L & P-plate training and certification) will work with all cars
- Open standards with automobiles allow for competition between suppliers, which is good for government buyers.

Another Example of Open Standards

- Computer (PC) hardware. How can you tell?
  - Hundreds of manufacturers.
  - Tens of thousands of system builders (from IBM, HP and Dell down.)
  - All of it is interoperable. One PC component will work with other PC components (of the same vintage/class,) from different manufacturers.
- Outcome? PC prices have dropped in (CPI-adjusted)dollars by a factor of 10 since 1994! They have simultaneously become 100 times more powerful! That's a 1000-fold price-performance improvement

What Happens When You Don't Insist on an Open Standard?

- You end up with Microsoft Platforms and Applications.
  - There is only one supplier (Microsoft)
  - There is no real attempt at interoperability; for example, making Microsoft Exchange work with other groupware clients or providing Visual Basic runtimes for non-Windows platforms.
  - Every effort is made to lock in customers (you!) into their one platform, in perpetuity.
- Outcome? Microsoft Windows + Microsoft Office have become twice as expensive over the past 10 years, with very little obvious increase in performance. This is a *two-thousand-fold degredation* in relative price-performance compared to the Open Standard oriented PC hardware sector.

Advantages of Open Standards

- Open protocol, document, platform and programming standards, which are of great tactical negotiating advantage to government
- They allow the agencies in question to switch vendors and shop around for different solution products and providers; this is almost impossible to do if an agency is locked into proprietary data formats and communications protocols.
- Open Standards are more likely to be in place when dealing with Open Source technologies, as all communications protocols and data formats are documented, within the code itself, if nowhere else.

Challenge of Deploying Open Standards

- Governments worldwide, like many other sectors, have often been caught out by adopting technology which isn't based on Open Standards.
- Platforms and tools are often chosen purely on short term requirements, not long term strategic goals.
- The psychology of buyers: non-Open Standards-based suppliers are often well-heeled and have exceptional abilities at marketing and selling even sub-standard software, making them seductively easy to select.
- Product end-of-life and migration costs are rarely included in procurement costings. This is a mistake!

Solution – Making Open Standards a No-Brainer

- Make open, documented protocols, formats and Application Programmer Interfaces a *must* for any government procurement of software, in much the same way that the military requires blueprints for military hardware acquisitions.
- Encourage a management culture where short term decisions on platforms are to be avoided and strategic benefits are always considered.
- Ensure that end-of-life and migration costs are *always* factored into any procurement decision.

Migration Costs – A Scenario

- Government agency wants to purchase a full-function word processor application for 500 users:
  - The current procurement process would only look at purchase and integration costs. The costs of migrating the users, templates and documents from this vendor/product to another in future, are not considered.
  - The vendor of that product line knows this, and capitalizes on it during the next round of pricing negotiations.
  - The new process would calculate a cost for such a end-of-life migration, and append it to the procurement costs.

Migration Cost - Detail

- The more *closed* a competing word-processor is, the more expensive to extract the documents invested into its care. Closed document formats are extremely difficult to transfer to other formats
- The more open (in document formats, macros, APIs) a word-processor is, the lower the end-of-life migration costs.
- If we factor this additional cost and substantial risk factor in, Open becomes a no-brainer.

The Challenge of Intellectual Property Ownership

- The essence of this argument goes something like this. We, the taxpayers, provide the money for everything that government does.
- Government pays some of that money for software development to occur, to meet its own needs. Should not the resulting output (software,) perhaps in a cleansed form, be made publicly available to those who essentially provided the funds for this activity?
- Australian governments understand the notion that publicly-funded endeavours, such as The Bureau of Meteorology, should make the bulk of their IP available to the public. Why not with software?

The Fruits of Government Funded Software - What Happened in Past

- The government would enter into some complex legal agreement whereby the government agency in question could gain some usage rights while a 3rd-party service provider tasked with developing the software gains distribution and profit rights.
- Some sequence of events will occur whereby, more often than not, this arrangement will falter, most likely to the detriment of the originating government agency and to the code base, resulting in an orphaned code base some years down the line.
- There *has* to be a better way. There is.

Commons IP Makes Common Sense: Releasing the Code!

- Most taxpayers would probably think this makes sense, but what does the government get out of undertaking such a code release and in what manner should the release occur? Let's look at a hypothetical scenario:
  - A government agency hires a solution provider to develop an implementation of a powerful new encryption algorithm, to be used in a VPN scenario linking government offices around the country.
  - As the government owns the intellectual property to the resulting technology, why should it consider releasing this code to the public?
  - For starters, there is the possibility of security fixes and enhancements which might arise. Since we are dealing with specialised cryptographic code, it stands to gain a great deal through 3rd party code auditing and sanity checking. (CSIRO, Universities)

Advantages To Opening the Code: The Fluoride Argument.

- Another advantage of broadly opening the code, is that it helps prevent code decay.
- Code decay occurs when a codebase isn't actively maintained by those who know and understand it.
- It happens when the original developers of a system move on to other projects, employers or clients. Years pass, libraries become obsolete, underlying platform components change and parts of the code stop working.
- It happens too often in both corporate and government sectors.

Leveraging the Global Brain: Open Source Innovation Networks

- If a technology platform is considered potentially of interest to a broader group of users and developers and is released into that group, it stands a much better chance of avoiding such decay as there are a larger number of interested developers maintaining that code, due to the larger group of parties which has a vested interest in keeping that code healthy.
- Furthermore, it creates a wider community of skilled practitioners, expert in that codebase (and available for hire!) and a far greater likelihood for substantially extending the longevity of the technology.
- This benefits the government, who can now continue using that original codebase for years longer than would have otherwise been the case. This saves IT budgets. It saves taxpayers' dollars.

But, Surely We Can Sell This IP? The Arthur Daley Syndrome

- Could the government agency in question gain the advantages of enhancing the longevity of its codebase by awarding the rights to ongoing development to a single commercial developer? Maybe make some money on this 'investment'?
- Unlikely. The contortions and twists which arise in many a commercial transaction may leave that government agency without any legal access to the updated code. As all of us in business have experienced, initial goodwill on projects such as this can quickly sour.
- The originating agency needs a way to protect its perpetual access to the source code. If the codebase had been released under an Open Source licence, the agency in question would have legal rights to that code in perpetuity.

Warranty and Indemnification Issues for Open Source Code Used by Government

- Among the issues that government intellectual property specialists are considering in their increasing interest in all matters Open Source, is product warranty.
- There has been some discussion that by obviating all warranties, the GPL and other Open Source licences may be contravening the Australian Trade Practices Act.
- There's nothing new in software licences obviating warranties. Most licences for proprietary software use very similar language in blanket eradication of warranties for end-users.

What Does the Alternative Offer?

- For example Microsoft's warranty for XP is clearly laid out in the EULA:
  - *YOUR EXCLUSIVE REMEDY. Microsoft's and its suppliers' entire liability and your exclusive remedy shall be, at Microsoft's option from time to time exercised subject to applicable law, (a) return of the price paid (if any) for the Product, or (b) repair or replacement of the Product, that does not meet this Limited Warranty and that is returned to Microsoft with a copy of your receipt.*
- In other words, if you paid no licence fee (as would be the case with Open Source software) you have no warranty.

How is Open Source different?

- Since declaiming or annulling warranties has been common practice for software companies for decades, why does it matter now with Open Source software?
- The answer is it doesn't. It appears that with the onrush of Open Source into government departments, the intellectual property people within governments have been taking a much longer and harder look at licencing of both proprietary and open technologies.
- Part of this is because they now have licences, such as the BSD and the GPL, which are fundamentally different from what was there beforehand.
- It is this difference which encourages consumers, even governments, to reconsider their options, because now they have some!

Caveat Emperor Penguin?

- Indemnification is the process by which one party provides certain protective assurances to another party.
- As with most proprietary software, Open Source comes with no indemnification. It's free, you use it at your own risk.
- This is disquieting for many government agency procurement staffers, but only those who didn't realise that they have been labouring, for years, under the same lack of licence indemnification from proprietary vendors, too.

GPL & Trade Practices Act: Really at Odds?

- It is not clear that there is a warranty implied in the supply of Open Source software
- This is due to the fact that the Act requires there to be a 'contract' (even implied) in place for the supply of goods and services.
- Such a contract is absent with an anonymous Internet download of Open Source Software.
- How should a government agency therefore protect itself with respect to warranties for Open Source?

The Solution: Common Government Contracts to the Rescue

- Federal and state agencies generally require service providers to execute such instruments as Endorsed Supplier Agreements, Common Use and Panel Contracts and the GITC4.
- These types of contracts generally indemnify the government purchaser and establish obligations and guarantee levels on the vendor/service provider.
- This should provide the missing piece of the puzzle in obviating perceived risk for acquiring Open Source solutions.

Warranty and Indemnification Issues for Open Source Code created by Government

- If a government agency adopts an Open Source solution, but needs it extended, what should it do with the additional code?
- There are great synergistic and re-use opportunities available by releasing the code.
- If the agency makes the solution available to other divisions, there's no real concern from a warranty and indemnification perspective, as it's just another arm of the same legal entity, the Crown.
- If, however, the agency wants to broadly release this source code, what are the implications?

We've Been Here Before

- There is existing precedence for such release and a reason to believe that it is safe to proceed.
- The federal government's Bureau of Meteorology accrues, and publishes information into the commons.
- There is potential for this information to somehow lead to the loss of property or life, raising legal liabilities.
- This doesn't mean that the Bureau should stop publishing
- It is implied and universally recognized that this information does not come with any indemnification or warranty. This is essentially the Good Samaritan argument.
- Bureau of Statistics, ABARE, CSIRO in same boat

The Solution: The Independent Code Clearinghouse

- Bureaucrats are very risk-averse individuals. What else can we do to relieve their sense anxiety about releasing code?
- The Australian government can create and safely release Open Source code by funneling it through a separate legal entity, a not-for-profit clearinghouse, assigning all legal copyrights and responsibilities which have been created, in the process.
- This clearinghouse will present a low-risk profile; it will never be wealthy enough to target legally.
- As the code will be Open Source, there will be a perpetual and un-encumbered legal access safeguard to the code, for the government.

Industry Development Advantages of the Clearinghouse Model

- Creating such a clearinghouse will also help funnel more capital-intensive intellectual property into the local ICT context, boosting the volume of quality code available for industry to re-use, helping all developers build better systems.
- Local developers would gain access to the code first, and understand the local environment and culture in which it was designed for, allowing them a time-to-market competitive advantage in the commercialisation of this Open Source code.

Vesting Copyright with the Service Provider

- One final approach to releasing code developed on behalf of the government, is to do it through a service provider/vendor.
- For any new project work or for any enhancement work on an Open Source application, the government agency vests copyrights and IP ownership/liability of that code with the selected solution provider, as long as the code was released under an Open Source licence; perhaps the fairly neutral LGPL.
- This removes indemnification and warranty concerns the government legal people may have.

Call to Action

- Standards Australia should begin work on ratifying which Open Standards should be adopted by government, preparing recommendations and outlines.
- Government must move towards a whole-lifecycle cost analysis of acquiring and end-of-lifing of software technology.
- Research should be funded into determining the cost efficiencies of a government-only CodeBank and a public Open Source Clearinghouse trust.
- Local Open Source service providers have to greatly increase their understanding of how government buys products and services and adapt to that process.

Questions?

Thanks for your time.

(These slides are available for your reuse.
Contact me for them.)