Australian Government
Department of Finance and Administration
Australian Government Information Management Office

A Guide to Open Source Software
for Australian Government Agencies

Developing and Executing an ICT Sourcing Strategy

Australian Government Information Management Office
a Business Group of the Department of Finance and Administration

Australian Government
Department of Finance and Administration
Australian Government Information Management Office

A Guide to Open Source Software
for Australian Government Agencies

Developing and Executing an ICT Sourcing Strategy

Department of Finance and Administration 1 74082 084 3
A Guide to Open SourceSoftware for Australian Government Agencies (print)

Department of Finance and Administration 1 74082 085 1
A Guide to Open Source Software for Australian Government Agencies (online)

Table of contents

Glossary **6**
Foreword **7**
Introduction **8**
Sourcing open source software 8
A definition of open source software 8
What is source code? 9
Open source software and commercial software 9
History and development of OSS 10
OSS usage within government 10
Overview of the OSS industry 11
Larger vendors 12
Established SME vendors 12
Boutique consultancies and SME vendors 13
The Australian open source industry 13

Typical concerns about open source software **14**
Cost of licences 14
Availability of support 14
In-house support 15
External support 15
Maximising flexibility in support options 16
Software reliability 16
Maturity and longevity 16

Sourcing open source software **18**
Sourcing scenarios 18
In-house sourcing 19
In-house procurement checklist 20
External sourcing 20
Open source products and single sourcing agreements 21
Incidental sourcing 22
Custom software development 22

Preparing a procurement plan **23**
Evaluating the business case 23
Defining business requirements and priorities 24
Sourcing issues to consider 25
Writing inclusive RFTs 25
Defining selection criteria 26
Assessing the value of OSS solutions 26
Meeting Australian Government requirements 27

Financial management and accountability 27
Procurement and best practice guidelines 27
Chief executive instructions (CEIs) 27
Endorsed Supplier Arrangement 28
Security management 28
Government Information Technology and Communications (GITC) contracting framework 28

Risk analysis and risk management 29
Availability of viable competitors 29
Warranties and indemnities 30
Lifecycle of open source products 30
Early phase: technology preview 31
Middle phase: early adoption 31
Maintenance phase: long-term support 31
Deciding when to adopt OSS products 32
Impact of access to source code 32
Assessing technology risks 33
Maturity and fitness for purpose 33
Analysis of technology roadmap 34

Risk mitigation procedures 35
Assessing sourcing risks 35
In-house sourcing 36
External sourcing 36
Assessing long-term product viability 37
Assessing technology roadmap risks 37
Managing risks in technology roadmap variation 38
Due diligence risk mitigation checklist 39

Understanding the legal context 40
Assessing licence risks 40
OSS licence types 41
Implications of open source copyright 41
Contrasting open source and proprietary licences 41
Contrasting different open source licences 42
Modification scenarios 42
Examining open source licences in detail 44
Applicability of open source licences in Australia 45
Trade Practices Act considerations 45
Broader intellectual property implications 45
Who owns and controls open source software 46

Sharing OSS solutions 47
Modifying products for use within one agency 47
Sharing modified products with other Australian Government agencies 48
Making modified products available to the open source community 49

Appendix A: Open source software resources 50

Appendix B: More about open source software 52
Open source software or free software? 52
Open source business models 52
Source code access: implications for agencies 53
Open source, open standards and open systems 54
Open source 54
Open standards 54
Industry standards and proprietary standards 55
Economic impact of open standards 55
Open systems 56

Appendix C: Open source software packaging 58

Glossary

AGIMO	Australian Government Information Management Office
Apache	A web server platform developed and released as open source software
BSD	Berkeley Software Distribution, a model licence for open source software developed by the University of California Berkeley; also a version of the Unix operating system that exists in both open source and proprietary variants
CMS	Content management system
DCITA	Department of Communications, Information Technology and the Arts
DAM	Demand Assessment Methodology
ESA	Endorsed Supplier Agreement
GNU	GNU's Not Unix, a project aimed at developing a complete Unix-style operating system based solely on free software
GPL	General Public Licence, a model licence for open source software
HTML	Hypertext Markup Language
ICT	Information and communications technology
ISV	Independent software vendor
Java	A programming language developed by Sun Microsystems
LGPL	Lesser General Public Licence, a model licence for open source software based on the GPL that permits linking to proprietary software
Linux	A computer operating system developed and released as open source software
Mozilla	A model licence for open source software; also an open source web browser
MySQL	A database system developed and released as open source software
OSI	Open Software Initiative
OSIA	Open Source Industry Australia, a broadly based group for local OSS vendors
OSS	Open source software
NOIE	National Office for the Information Economy
Samba	An Australian-based open source software initiative providing interoperability between computers running Linux/Unix and those running Windows
SME	Small to medium enterprise
SQL	Structured query language: a common technique for querying databases
TCP/IP	Transmission Control Protocol/Internet Protocol, the basic software protocol for Internet communications
VAM	Value Assessment Methodology
W3C	World Wide Web Consortium

Foreword

SENATOR THE HON ERIC ABETZ
Special Minister of State

The increasing maturity of open source software and open source platforms offers significant potential benefits to the Australian Government and the wider community. Open source software development, using open standards, can support greater interoperability between systems and enable system sharing. It can offer original solutions to problems not addressed by proprietary software and it has the potential to lead to significant savings in Government expenditure on information and communications technology (ICT).

Open source software is already in wide use within Australian Government agencies, and is particularly well established in ICT infrastructure support and management systems. Open source solutions are being offered by a variety of vendors ranging from small to medium-sized Australian companies to large multi-nationals. There is a growing market for companies that implement and support open source solutions in business and government.

Many agencies are now looking to explore the possible wider application of open source software within their ICT systems. Under the Australian Government's procurement framework, "value for money" is the key criterion against which competing fit for purpose solutions should be assessed. All solutions - open source or proprietary - which can meet an agency's functional specifications should be considered by an agency when it is undertaking software procurement. However the different nature of open source software can make such assessments, and comparisons between different solutions, difficult.

This Guide seeks to assist agencies by providing practical information and approaches for agencies to consider when assessing open source solutions. Risk management procedures and the different contractual considerations that can apply to open source software issues are addressed in the Guide, as are cost of ownership issues. Understanding cost of ownership issues for open source software is important because, under an open source model, costs are incurred at different phases of the implementation and operation of an information technology system.

The Guide is a companion document to the 2004 publication *A Guide to ICT Sourcing*. Between them these publications provide a basis for better ICT procurement decisions across the whole of the Australian Government.

April 2005

Introduction

This guide is a complementary document to *A Guide to ICT Sourcing* published by the Australian Government Information Management Office (AGIMO) in May 2004.

The purpose of this guide is to provide Australian Government agencies with background information and processes to better understand, analyse, plan for and deploy open source software (OSS) solutions in appropriate situations. It is not intended to direct government users and agencies towards specifying open source technologies as part of their procurement practices. Decisions on the adoption of open source software should be made using the standard Australian Government criteria of fitness for purpose and value for money. See 'Preparing a procurement plan', page 23, for more detail.

Sourcing open source software

Open source software has various attributes that differentiate it from the traditional software procured by Australian Government agencies. These differences can be easily identified and described to enable agencies to better understand what changes (if any) to sourcing procedures may be required to accommodate OSS solutions.

In many respects, sourcing OSS is little different to sourcing traditional proprietary software (otherwise known as commercial or off-the-shelf software). In general, the processes and strategies defined in *A Guide to ICT Sourcing* apply to decisions about OSS. Where there are additional processes or differences in strategy, these are covered within this guide.

As open source procurement is relatively new to government, this guide provides background and explanatory material to assist agencies in better understanding the risks and opportunities involved in sourcing OSS. In addition, the appendices offer detailed information about OSS, including a detailed comparison of 'open source', 'open standards' and 'open systems'.

A definition of open source software

Open source software is a type of computer software defined by several specific attributes that relate to its licensing and legal framework. Often it also involves a distinctive development and distribution model.

At present, the primary arbiter of what constitutes open source software is the Open Source Initiative[1]. The Initiative sets out various rights and obligations for developers, distributors and users of OSS. These rules define the basic licence conditions under which software must be released to be considered 'open source'. These licence conditions give the users of OSS the right to:

- Use the software for any purpose;
- Make copies of the software for any purpose;
- Access or modify the source code of the software for any purpose; and
- Without payment of a royalty or other fee, distribute copies of:
  - the software (including distributing the software as part of an aggregate distribution containing software from several different sources); or
  - a derived or modified form of the software (either in compiled form or as source code), under the same terms as the licence applying to the software.

Source: :www.opensource.org/documents/definition.php

What is source code?

All software is written in what is known as source code. This refers to the underlying, human-readable programming instructions produced by software developers. In most circumstances, these programming instructions are compiled into what is known as binary or machine-readable code; this is the code that actually 'runs' or 'executes' on a computer.

A significant difference between proprietary software and open source software is that OSS developers make the source code available to anyone who wants access to it. In contrast, proprietary software vendors normally only release their products in binary form. In many circumstances, agency users do not need access to this source code, nor should they need to modify it for their specific requirements. Access to source code is, however, valuable to agencies by virtue of the economic flow-on effects that accrue when multiple vendors offer competing products based on the same technology. Access to source code also reduces the risk of vendor lock-in.

Open source software and commercial software

Open source software is often considered to be 'non-commercial'. This is not necessarily the case. Most OSS used by government and industry is available under commercial terms for commercial purposes. It just happens to be released under a licensing scheme that allows free redistribution.

Open source software is available from a wide variety of commercial suppliers. This includes many vendors that also supply software and solutions based on proprietary software. Open source licences do not preclude commercial exploitation of the software.

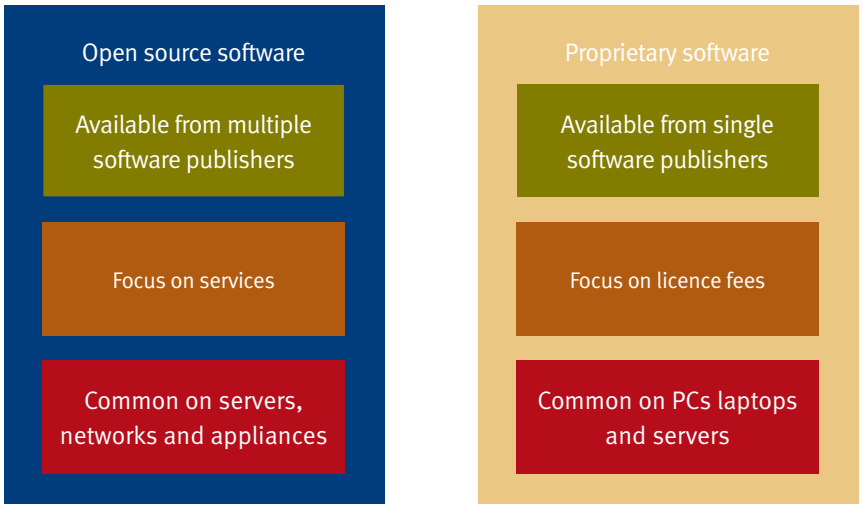
In most circumstances, users do not pay software licence fees for open source software. However, commercial OSS vendors deliver open source products through a model where fees are charged for services rather than licences. See Appendix B, page 52, for further discussion of these issues.

Most open source software is also copyrighted by its author(s). However, open source licensing requirements give users additional rights and obligations, including the right to reproduce and redistribute the software.

The general differences between OSS and proprietary software are set out in Figure 1.

[1] Open Source Initiative: :www.opensource.org.

FIGURE 1. DIFFERENCES BETWEEN OSS AND PROPRIETARY SOFTWARE



History and development of OSS

For many years, there has been a general trend in the information and communications technology (ICT) industry towards increasing openness of platforms, communication protocols and data storage formats. This trend has encouraged the rise of open systems technologies based on open standards. Any vendor that wished to participate could develop solutions based on these open standards (although sometimes they needed to pay for access to the standards). See Appendix B, "More about open source software", page 52, for further discussion of these issues.

Open source software has existed in some form for over 20 years. However, it was only during the last five years that OSS technologies were widely adopted by the ICT industry.

In recent years, open source software has attracted strong interest from government agencies, the private sector, the ICT industry and independent software developers. OSS products have a number of unique characteristics that appeal to each of these constituencies.

OSS usage within government

Major ICT users in both the private and public sectors have used OSS products and solutions in specific areas for many years – with or without the 'open source' label attached. The use of OSS is particularly widespread in areas such as network infrastructure, single-purpose computer servers, security, Internet and intranet applications and network communications. At present, OSS tools are less widely used on computer workstations, laptops and desktop personal computers (PCs).

Increasingly, open source is expanding into other areas of ICT, both niche and mainstream. Contexts where OSS solutions are emerging as a common choice include:

- ***Network infrastructure***: including software for domain name service (DNS), IP address allocation (DHCP), web services, application services, proxy servers, directories (LDAP), packet shaping and communications optimisation;
- ***Database servers***: prominent open source database servers include Firebird SQL (formerly Interbase), Ingres, MaxDB (formerly Adabas), MySQL and PostgreSQL. In addition, many proprietary database servers are now available on open source operating systems;
- ***Security systems***: including firewalls, intrusion detection and analysis, honeypots, IPSEC and other virtual private network (VPN) systems, packet-sniffing and analysis, antivirus software and anti-spam filtering;

- ***Internet and intranet publishing***: including web servers, content management system (CMS) platforms and workflow management tools;
- ***Document management***: including automatic electronic document capture systems, revision management systems, data capture technologies and archiving systems;
- ***Email and communications***: including numerous solutions for email, general groupware (group calendaring, shared address books, reminders, public folders) and instant messaging servers;
- ***Application servers***: including widely used web application servers based on PHP, Perl, Python and ZOPE scripting tools, Java and Java 2 Enterprise Edition (J2EE) servers such as JBOSS and the Mono and dotGNU .NET open source application servers. In addition, many proprietary application servers now run on open source operating systems;
- ***File and print servers***: tools covering most major file sharing protocols, such as Unix NFS, Microsoft SMB/CIFS and Novell Netware NCP;
- ***Storage***: several network-attached storage appliances are built primarily on open source platforms;
- ***Limited-function workstations***: fixed-use workstations that provide basic web, email, terminal access and office productivity functionality for call centres, kiosks and similar uses;
- ***High-performance computing***: this includes single-image systems with multiple microprocessors (vertical scaling), clusters based on large numbers of low-cost systems (horizontal scaling) and other types of supercomputers; and
- ***High-performance technical workstations***: including multi-processor, 64-bit and large-memory systems for computation-intensive applications such as scientific analysis, meteorology, modelling, 3D computer-generated imagery (CGI) and video-processing functions.

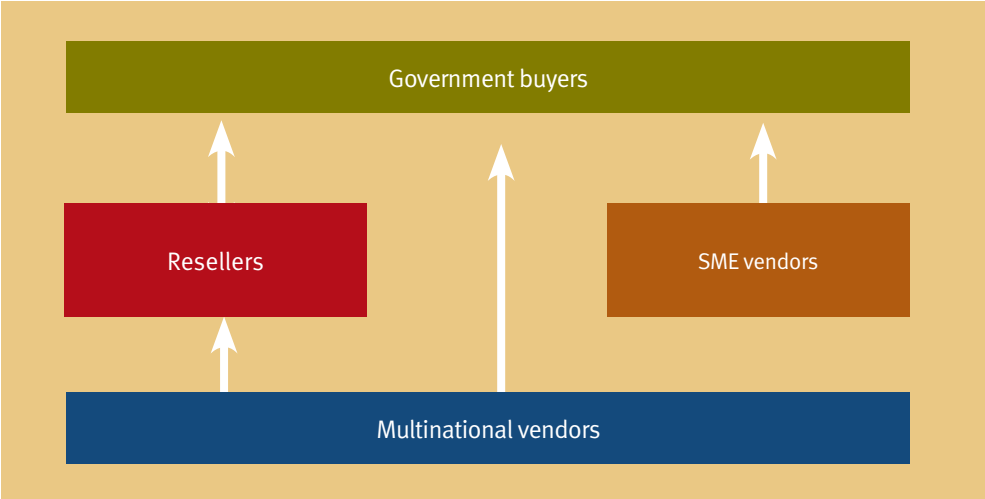
Each agency must determine where open source software may have a role to play according to its own context and priorities.

Overview of the OSS industry

As open source platforms have entered the mainstream, larger independent software vendor (ISV) organisations have begun producing versions of their enterprise applications or systems technology that run on these platforms. Major outsourcing solution providers have also extended their support services to cover common open source software and platforms. However, the market for OSS solutions and services still tends to be dominated by smaller vendors.

Figure 2 provides a basic overview of the most common components of the OSS industry.

FIGURE 2. OSS INDUSTRY STRUCTURE



Larger vendors

One group of solution providers in the OSS space includes large, well-established hardware and software vendors as well as the largest 'pure-play' open source companies. The Australian Government has a track record of dealing with these larger vendors because of their national capacity and stable trading histories.

Established SME vendors

A second group consists of the more mature small-to-medium enterprises (SMEs) operating in the open source space. Many of these are Australian-based companies. In addition, numerous mid-sized hardware and software vendors now offer open source solutions. These vendors often have extensive experience in dealing with government requirements.

The smaller players in this group include service providers and product vendors. Many offer broad levels of expertise across a range of open source technologies. Many have added their own functionality to open source software to create enhanced products. These products are delivered in the form of either targeted bundles of software and services (for example, content management systems, business intelligence software, line-of-business applications) or more general hardware-software bundles (for example, departmental groupware servers, firewalls, hardened gateways for wireless authentication and other appliances).

Established SMEs generally have enough experience in dealing with public sector organisations (often at a state or local level) to make procurement of solutions reasonably straightforward.

Boutique consultancies and SME vendors

The last group of vendors in the open source industry are smaller, specialist players. These organisations are predominantly focused on specific market segments or utilise a cluster of related open source technologies. These vendors may offer superior expertise in a nominated open source product. If an agency is interested in adopting that product, perhaps through in-house acquisition, it may be advantageous to formulate a process to obtain technical support from a specialist vendor.

The Australian open source industry

Australia's OSS industry is still in a formative phase. At present, there are an estimated 300 to 400 local small-to-medium solution providers that specialise in open source software[2].

The majority (over 90%) of these are smaller players with less than five staff. A handful are slightly larger (around 30 staff) while none have more than 100 staff at the time of publication. These vendors are also geographically localised, offering points of presence and support around specific state capitals or regional centres. Very few have national presence at this stage.

A majority of these firms have been in business for less than five years. Few have been in business for more than ten years.

[2] Data in this section is drawn from analysis of firms who are members of Open Source Industry Australia, a broadly based industry group. See OSIA website: :www.osia.net.au.

Typical concerns about open source software

Formal procurement of open source software is a new undertaking for many government agencies. As a result, some agencies have concerns about open source software, including licence costs, support options, reliability and maturity. This section addresses these concerns.

Cost of licences

Most open source software does not have associated licence fees. Vendors generate revenue by providing implementation and support services.

Some vendors sell solutions that use a combination of open source and proprietary software sold in a licensed bundle. These solutions are typically priced on a per-seat or per-server basis.

Some vendors bundle open source products with service-level support agreements. Agencies interested in procuring such products need to assess the value of this kind of bundling. It may be more cost effective to acquire the underlying product from one vendor then negotiate ongoing service-level support agreements separately using a selective sourcing process. See Appendix B, page 52, for further discussion of these issues.

Availability of support

The most frequent question associated with the procurement of open source software centres on support. Specifically, who will support the agency if there is a problem with the software? Agencies need to understand this issue to conduct an appropriate assessment of the risks involved with various sourcing options.

At the base level, all open source software is maintained and supported by a user community. This support is ad hoc and there are no service-level guarantees. However, experience has shown that one or more vendors quickly moves to support a major OSS product if a market is shown to exist.

Support for OSS products is generally offered in a manner similar to proprietary software. Most vendors and resellers offer service-level agreements, on-call helpdesk services and the purchase of support packages. Australia has several hundred small and mid-sized vendors offering support to industry and government under such commercial terms.

OSS products with widespread usage tend to have a broader choice of support services and providers. Popular technologies like Linux, Apache, MySQL and Samba generally enjoy global support coverage from some of the largest vendors in the ICT industry. Support terms for these products are generally similar to the commercial support terms these vendors offer for proprietary software.

In-house support

An agency may opt to in-source the procurement and support of an OSS solution when its in-house technical staff have the necessary skills to deploy and manage the solution on a day-by-day basis. However, for risk mitigation purposes it is important that the agency be aware of any limitations of its in-house resources. The agency may choose to formulate plans to procure expert external support if needed.

It is important to establish contingency plans before permitting the solution to move to the operational (production) phase. Ensuring such plans are in place can reduce the severity and duration of any problems that might arise. In some circumstances, contingency plans may provide the risk mitigation strategy an agency needs to actually proceed with in-house sourcing.

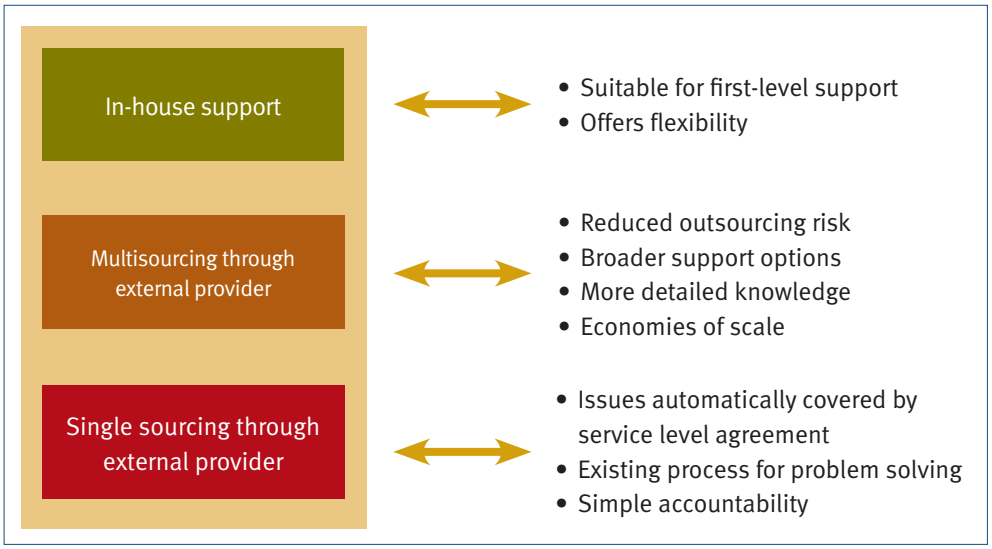
External support providers are available for most widely used open source products. Agencies should conduct a detailed analysis of the viable vendors offering support for products of interest as part of the initial procurement process.

External support

For agencies that opt to outsource the support of an OSS solution, technical support is normally the responsibility of the selected vendor. The vendor should provide first and second-level support. It should also undertake the necessary collaboration and correspondence with the developer community that created the software to ensure resolution of third-level (bug fix) issues. From the agency's perspective, the entire process should be similar to engaging proprietary software vendors.

Figure 3 shows the strengths and weakness of different support sourcing options.

FIGURE 3. IN-HOUSE AND EXTERNAL SUPPORT OPTIONS FOR OSS



Maximising flexibility in support options

Open source software offers considerable flexibility in its support options by virtue of its development, release, distribution and licensing regimes. Many popular OSS products have a broad spectrum of support offerings from a range of competing vendors. Each vendor may offer clients support with helpdesk, troubleshooting and bug-fix services.

This multi-point distribution model is a key difference between open source software and traditional software. Flexible support options can give agencies latitude to achieve better value in the procurement of open source solutions.

Software reliability

Another common question about OSS products relates to the issue of quality. Given that most OSS products are available free of licence costs, some agencies question if the software can be considered reliable.

Just as in the proprietary software world, there is an enormous variety of open source software available. The major online repositories list around 70,000 OSS packages. To date, there have been no definitive studies reviewing this software against quality and reliability metrics. It is therefore difficult to make broad statements about the robustness of open source software.

Most open source software fits into particular categories. Some prominent categories are database engines, scripting languages, application servers, content management systems, office suites and desktop groupware. Within any one category, there are sometimes several hundred products available. Many of these products have not yet matured to desirable levels of quality and reliability and are therefore inappropriate for consideration by Australian Government agencies. Other OSS packages are mature, stable, functional and widely used.

Maturity and longevity

The collaborative model used for most open source projects involves a feedback-based development model where maturity and reliability improve as usage increases. During the initial stages of a product's lifecycle, open source applications can be less robust and less reliable than released versions of proprietary software. Many open source development communities follow a release philosophy known as 'release early, release often'. Early versions of these products are normally only intended for limited adoption, so agencies should be cautious about considering such products for important functions.

If properly designed and properly managed, open source applications can develop with remarkable speed. OSS projects that meet market demand tend to accrue additional developers, vendors and early-adopter technologists quite quickly. However, poorly designed or managed projects can wither and make little progress over time. Applications without support from a strong developer community are best avoided.

The more successful projects accumulate a growing community of developers and users. In turn, the software codebase, documentation, website, mailing list support and discussion forums develop. This momentum tends to attract additional attention, resources, developers and users to the project.

During a project's first dozen or so 'release early, release often' iterations, the software may be still too immature for production use. However, at this stage it can reach a state where agencies find enough benefit to warrant further investigation. Properly cultivated, this growth phase could enable the software to mature sufficiently to find mainstream acceptance.

In some ways, the success of one open source software product over another mimics the selection process at work in natural evolution. The strongest, most viable projects accumulate the scarce resources (early adopters, developers, testers) that allow development to further accelerate towards maturity. This in turn attracts additional users, which accelerates the maturation process further.

Users and developers generally benefit from selecting a strong competitor, as this has the greatest chance to become a viable platform with a large user community. This significantly increases the likelihood of product longevity.

This factor has particular importance for government agencies. Selecting a product that may struggle to evolve to maturity involves serious risks. It is therefore important to undertake an analysis of the open source product to determine its current level of maturity and success. Not all OSS products need to be the top of the evolutionary ladder to be viable for selection. However, selecting a product with strong support from developers and users can reduce risk.

In most OSS technology categories, there is generally more than one strong competitor. This is particularly likely in categories that can be broken down into different ecosystems or niches. A computer operating system is the best example: its usage is so diverse that different OSS projects can succeed simultaneously by targeting a specific scale (mobile devices, desktop systems, servers) or a particular technology model (distributed computing, grid computing, monolithic servers).

Another example is SQL database engines, where there are at least a dozen viable open source projects. Many of these projects target specific niches such as embedded devices or enterprise back-end servers. The market also has sufficient size and breadth to support numerous SQL database engines. Within a particular niche, each product may have different strengths and weaknesses, so agencies should evaluate any potential products carefully.

Sourcing open source software

For most agencies, procurement of open source software is a new endeavour. This means many agencies will approach OSS cautiously. Under most circumstances, however, there is little reason to handle OSS procurement differently to proprietary software. Where differences arise, this guide highlights them and outlines their relationship with the existing procurement framework set out in *A Guide to ICT Sourcing*. Most of these differences are easily defined and managed.

Sourcing scenarios

There are four principal avenues for the introduction of open source software within government. These mirror equivalent procurement processes that exist for proprietary software, with one addition. These avenues are:

- ***In-house sourcing***: direct procurement of open source software, using in-house skills for sourcing and deployment;
- ***External sourcing***: using one or more external solution providers to deliver and deploy OSS-based solutions;
- ***Incidental sourcing***: deploying open source software as part of a larger sourcing solution, where the open source component is not the primary emphasis; and
- ***Custom software development***: the government agency develops or modifies open source software by itself.

These options are comparable to processes that most agencies already use. For example, many agencies already source ICT solutions through external service providers using either single-source or multiple-source arrangements. Most OSS platforms can be procured in the same way. In-house development using OSS technologies is analogous to customised development initiatives that some agencies currently undertake and the process can be managed similarly.

In contrast, direct sourcing is a new option that has no common equivalent in the proprietary software context. Many agencies find it more convenient to source OSS solutions directly using in-house technical expertise. Direct, in-house sourcing is not generally available as an option with proprietary software, which must normally be acquired from the vendor or an authorised distributor or reseller.

Let's consider these options in greater detail.

In-house sourcing

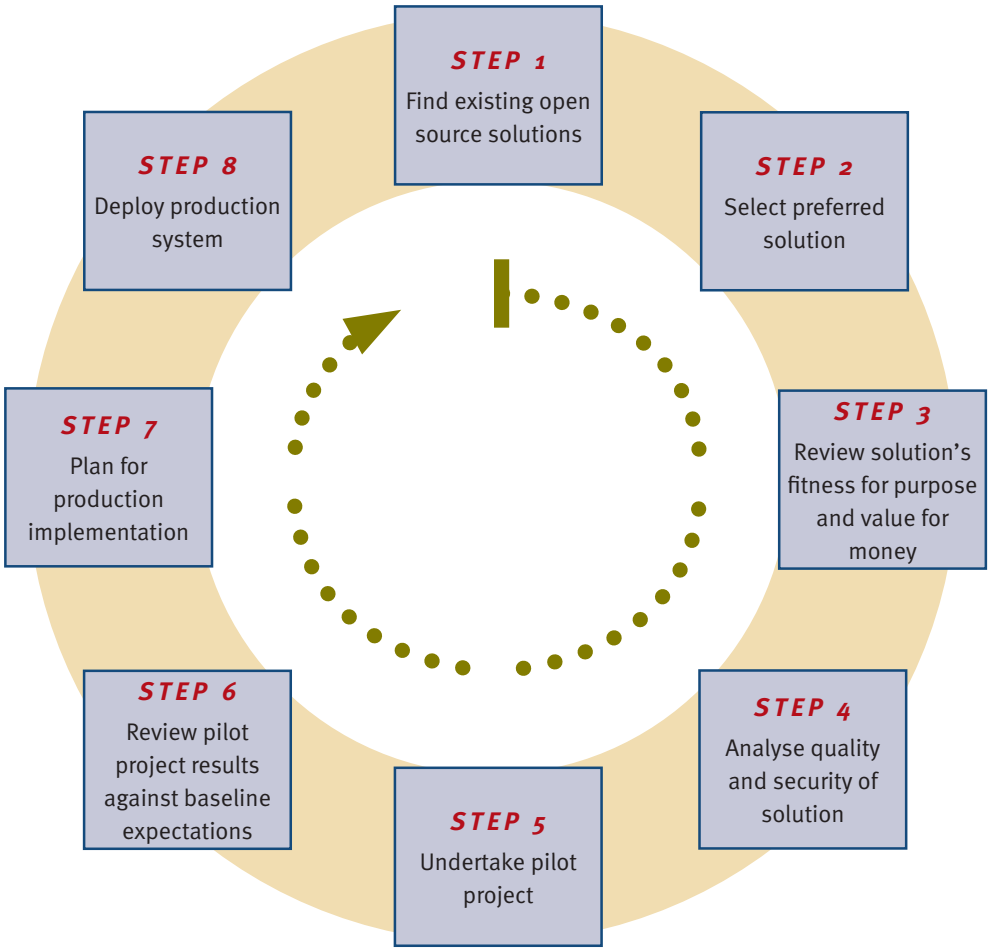
If an agency has the requisite skills in-house and flexible software procurement policies, it can obtain, install and use a substantial number of OSS products that are directly available from various online repositories. Browsing or searching these repositories allows agencies to find and download OSS products to meet their needs. Appendix A, "Open source software resources", page 50, lists many of these OSS repositories.

Such ad hoc software procurement is not normally available for proprietary software as this type of software is generally not available for free download. Where proprietary software is available for free download, there are often restrictions on its functionality or the duration of its use.

Direct in-house sourcing enables agencies to bypass formal procurement processes, purchase orders and expense requisitions; the principal costs relate to bandwidth usage and staff time. However, agencies still need to follow the standard risk assessment and change management procedures required by each agency's policy. All directly sourced OSS products should be downloaded, installed and tested by appropriately skilled technical staff before their use is approved.

Figure 4 sets out a model workflow for in-house sourcing of OSS products.

FIGURE 4. WORKFLOW FOR IN-HOUSE SOURCING OF OSS PRODUCTS



In-house procurement checklist

When undertaking in-house sourcing of open source software, there are a number of broad considerations for agencies to investigate before proceeding with software selection.

If a specific open source product appears to meet an agency's requirements and that agency decides to undertake a more detailed review, the following checklist provides a useful analytical tool to guide the assessment process.

In-house procurement checklist	
Does the software run on operating system platforms used within the agency?	Yes/No
Does the software require any additional system components, libraries or modules that the agency needs to obtain, test and deploy on these existing operating system platforms?	Yes/No
Does the software have a clearly defined and easy to understand installation procedure? (Some OSS packages come in formats that are difficult for non-technical users to install.)	Yes/No
Does the agency have the in-house expertise to install, deploy and test the OSS system to determine its fitness for purpose?	Yes/No
Does the software have a clearly defined uninstall procedure? Some OSS packages do not come with automated uninstall utilities or scripts. However, open source software that comes without uninstall tools can generally be removed by deleting the directory it is installed into. However, this may need appropriate in-house technical skill levels.	Yes/No

External sourcing

Many agencies may not have the in-house technical capabilities to source OSS products directly. As a result, they may prefer to acquire open source solutions through external service providers.

Most popular open source products can be procured via commercial solution providers. These range from large established vendors through to small-to-medium enterprises offering tailored solutions. However, there is normally a subtle difference between external acquisition of OSS and proprietary software. When an agency acquires an OSS-based solution through an external service provider, it is generally purchasing services and receiving the related software free. This differs from the proprietary model where the agency purchases software licences and receives support as a value-added service.

This difference is potentially confusing for agencies who have become familiar with the existing regime for proprietary software. It is perhaps the core difference between the two forms of software from a procurement perspective, so it is worth taking the time to understand what this difference means.

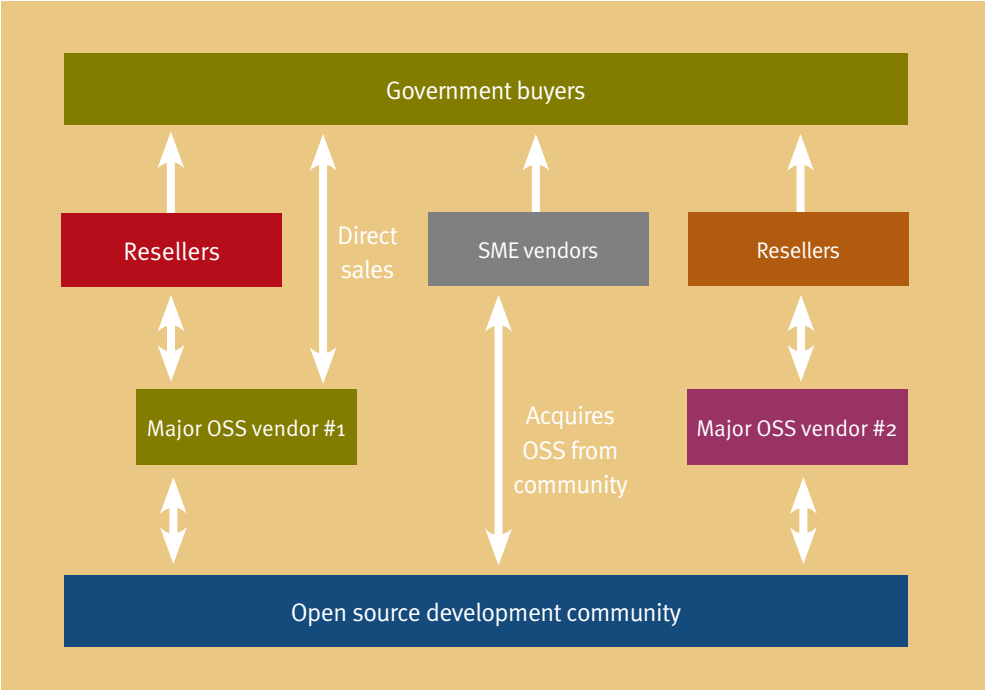
The first implication is that it is possible to acquire a solution built on open source software from a vendor that is not the originator or creator of the software. However, this vendor usually has the wherewithal to offer the solution as if it were its own. This differs markedly from traditional arrangements where various authorised resellers provide the same proprietary product from one software publisher but face limitations in the way they can support, modify, extend or maintain the product. Agencies often need to acquire these after-market services directly from the software publisher.

In contrast, any particular OSS product may have multiple software publishers that sell essentially the same product through multiple resellers. Each open source vendor should be able to offer the full spectrum of pre-sales and post-sales technical services on that product.

This arrangement has obvious repercussions for risk assessment. While no single software publisher or reseller can claim total ownership of a product (due to the open source licence conditions), each product has multiple independent development and procurement systems to support it. Therefore access to the product is secure and agencies have a strong bargaining position. Vendors and resellers must demonstrate their ability to fulfil the service and support requirements that client agency requires.

Figure 5 shows some of the typical reseller arrangements that can arise with OSS products.

FIGURE 5. OSS VENDOR AND RESELLER ARRANGEMENTS



Once an agency has decided that an OSS technology solution fits its needs, the next step is to select an appropriate vendor to undertake implementation and ongoing support of that product. *A Guide to ICT Sourcing* lists the key criteria by which a vendor can be selected from a group of possible contenders. As part of vendor due diligence, agencies should evaluate the financial strength, stability and technical capability of the shortlisted vendors. In essence, the same due diligence rules apply for vendors and resellers offering either OSS or proprietary software solutions.

In addition, the availability of multiple vendors for any given OSS product reduces the risk of an agency finding itself left with an orphaned product selection. This multiple source attribute offers additional flexibility to agencies mapping out transition strategies for moving from one supplier to another at the end of an agreement.

Open source products and single sourcing agreements

As more open source technology is used in the ICT industry, existing service providers working in the government arena may introduce OSS solutions into client agency environments. This may take place as part of the mix of software used by a single source

service provider to fulfil its service level agreements with the client agency. Alternatively, OSS may be deployed as part of a new technical methodology or approach that the service provider may adopt.

In such cases, the vendor and the client agency need to address all matters relating to the proper operation, warranties and indemnities associated with this new software within the existing contractual framework. These contractual issues should be resolved in accordance with the standard contracts and *Financial Management and Accountability Act* (FMA) and Regulations. See page 27 'Meeting Australian Government requirements'.

Incidental sourcing

Historically, adoption of open source software within organisations has generally occurred on an ad hoc basis. Often OSS products were used as an incidental part of larger ICT projects. This happened for several reasons. Firstly, most open source software is easily available: it can be downloaded directly with no licence costs or other upfront fees. Many OSS tools are also platform-neutral and designed to be modular in order to increase their usefulness and applicability. These attributes make OSS tools easy to download and deploy for specific functions within larger projects.

Typical ad hoc deployment scenarios include:

- Web-based content management systems for in-house use;
- Edge-of-network infrastructure components;
- Perimeter defence and firewall systems; and
- Specialist technological or scientific uses.

Within government, OSS was often adopted by agencies that had in-house technical expertise. Technical staff were able to research open source software options then download, install and trial the product.

On the other hand, many agencies were not in a position to adopt this type of process, often because they lacked sufficient in-house technical expertise. However, the open source market is maturing quickly and there is now a broad spectrum of vendors offering OSS-based solutions, products and services that enable these agencies to contemplate open source solutions through external sourcing arrangements similar to those used for other technology products.

Custom software development

In-house development using OSS technologies is analogous to customised development initiatives that some agencies currently undertake and the process can be managed similarly. However, there is a significant difference between custom development with OSS tools and customisation of proprietary software. The most important differences relate to the potential to redistribute the OSS-based solution and the rights and obligations imposed on an OSS user with respect to the release of source code for any custom modifications.

Licensing issues are discussed in detail in the chapter on 'Understanding the legal context', page 40. Redistribution issues are covered in the chapter on 'Sharing OSS solutions', page 47.

Preparing a procurement plan

As open source has matured and become more sophisticated, so has the procurement process. Current industry best practice suggests that an organisation should, as far as possible, procure OSS tools with the same processes and mechanisms it uses for other ICT solutions.

Evaluating the business case

Open source software is now broadly accepted and supported by many major ICT vendors. This broad acceptance means the presence of OSS is likely to grow in many areas of the public sector in coming years. Mainstream adoption of open source software makes it possible for more agencies to safely consider its adoption.

This guide is not intended to either advocate or reject OSS products. However, for agencies considering OSS solutions on the merits, this section provides advice on issues the agency should consider when evaluating the business case for OSS solutions. Where necessary, it highlights differences between open source and proprietary software. This is designed to assist agencies to quickly understand the unique value proposition of both types of software.

Decisions about open source software should be made according to the same metrics and decision-making processes that are used for other ICT solutions. The primary considerations are fitness for purpose and value for money. Agencies should consider open source software on its merits, disregarding any industry fads or novelty value.

One issue to consider is the perceived viability of OSS technology. As OSS has a relatively new presence in industry and government, some agencies may be apprehensive about considering OSS-based solutions. To ameliorate such concerns, it should be noted that many government computer systems already run some form of open source software. For example, many agencies already use server, router and firewall systems based on open source implementations of the TCP/IP protocol for Internet communications. Furthermore, many transactions that occur through the Internet do so on open source platforms such as Apache, Linux and MySQL. See page 14, 'Typical concerns about OSS software' for further discussion of this issue.

Some agencies may have a natural affinity for open source software and they are likely to make a stronger business case for OSS products. For example, agencies that have used Unix computer operating systems for many years may feel comfortable using Linux, an OSS

operating system that shares many characteristics with Unix. In contrast, other agencies may not have the technical expertise and comfort levels to make the same business case.

As part of the process of determining the business case for a particular ICT project, agencies need to define core project attributes and other imperatives for consideration. This includes titles, descriptions, scope of target usage group(s) and project plan mechanics such as test and launch timetables.

After articulating objectives for the procurement of an ICT solution, agencies can review some of the attributes that may be intrinsic to OSS products to determine their applicability. The following scenarios illustrate this process in action.

Procurement scenario examples

Increasing longevity of document accessibility

In this example, an agency has a business objective to maintain long-term access to its electronic documents. The agency may mandate the adoption of open standards formats to be used for the long-term archiving of documents. This mandate may be defensible if the specific requirement is defined as: "We want to ensure we can retrieve all archived documents in the medium to long term."

A document format based on open standards is selected for this purpose. If an open source solution can be shown to implement the standards-based document format, then it can potentially deliver the mandatory outcome. The OSS-based solution can therefore be considered for procurement.

Maximising network interoperability

Let's consider another example, this time relating the adoption of network communications protocols. An agency launches a project with the objective of reducing interoperability problems between products from different vendors. The desired outcome is proper system-wide operation among disparate components for the full operational lifecycle of the procurement.

The project presents several risks. Unless vendors adhere to well-defined, open industry standards their communications protocols may diverge in future product releases. This could cause considerable compatibility and security problems for the agency and has the potential to lock the agency into a particular vendor's technology. However, if there is at least one viable open source contender, which adheres to open standards, the agency could deploy this to ensure the ongoing standards compatibility of the network environment. In this case, the OSS solution enables the agency to fulfil the project's risk mitigation requirements.

Defining business requirements and priorities

In many aspects of the procurement process, open source solutions are little different to proprietary software. The procedures and decisions that an agency buyer needs to undertake are often similar or identical. This is certainly the case when it comes to defining business requirements and priorities for an ICT solution.

A Guide to ICT Sourcing provides an overview of the principal issues agencies need to consider when procuring solutions. Refer to that guide for a complete explanation of these issues; this section outlines some of the most important elements of the requirements phase. In most circumstances, the top priority is continuity in the agency's business processes: ensuring the continuous operation of computer and network systems that an agency requires to deliver services to citizens and government.

Other factors that an agency might take into account when considering a new ICT system include:

- Making business processes more efficient and simple;
- Lowering technology costs;
- Simplifying ICT tools, processes and operations;
- Rationalisation of ICT platforms and systems;
- Standardisation of ICT platforms and systems;
- Decreasing risk exposure;
- Decreasing exposure to security issues;
- Making the agency's business platforms more robust;
- Reducing the impact of system failures; and
- Increasing platform diversification to reduce the risks of systemic failure.

Sourcing issues to consider

Agencies need to address a number of issues before they can make decisions about sourcing options in projects where open source software is judged to offer a viable solution. Some of these questions apply only to in-house sourcing scenarios, while others are applicable across all sourcing scenarios. Issues that agencies need to consider include:

- The agency's level of in-house expertise and comfort with the technologies involved;
- The complexity of migrating from existing platforms or applications to an OSS equivalent;
- The cost and complexity of changes required to data, systems integration, network protocols and document formats; and
- The level of re-training needed for staff to use the new solution.

These issues are not unique to open source solutions; many apply to any ICT project that introduces changes to established business procedures. These issues are only part of a larger migration process. *A Guide to ICT Sourcing* recommends that agencies establish a transition or termination strategy for any sourcing initiative. This should cover service-level agreements, contract governance arrangements, transition of IT resources and other salient issues.

Writing inclusive RFTs

Request for tender (RFT) documents are an essential part of sourcing ICT solutions from external vendors. Agencies need to take care to avoid introducing unintentional barriers that may discourage or inhibit open source vendors and resellers from submitting responses. These barriers normally take the form of specifying proprietary products by name or stipulating interoperability requirements that are not based on open standards.

To ensure RFT documents do not exclude any viable products offered by vendors, we suggest the following simple rules:

- Avoid specifying products by name (for example, "the solution must be delivered using product x");
  - *Where possible, specify what interoperability requirements are required (for example, "must be able to read documents in this format" or "must be able to share files with these products");*

- Where possible, avoid specifying brand keywords or trademarks (for example, "must offer BrandName® thumb-scanning");
  - *Instead, articulate generic attributes and functions as part of your requirements (for example, "must be able to work with standards-based thumb-scanning hardware"); and*
- Avoid specifying proprietary or exclusionary standards where possible. See page 52, Appendix B for more information on open standards;
  - *Where possible, specify open and vendor-neutral standards.*

Defining selection criteria

A Guide to ICT Sourcing provides a comprehensive overview of the typical selection criteria that agencies need to incorporate and consider as part of their procurement plan. The issues covered include:

- Vendor's ability to deliver the service;
- Establishing service levels together with metrics and processes for assessing services;
- Transition plan;
- Cost proposal;
- Compliance with bidding process;
- Quality of proposed vendor staff; and
- Vendor profile.

The selection criteria established by *A Guide to ICT Sourcing* apply equally when sourcing OSS. Agencies should consider each potential supplier's relative maturity and credentials to deliver the proposed solution. As many agencies lack experience in procuring open source solutions, the next section presents a brief overview of the OSS market.

Assessing the value of OSS solutions

To assist agencies in assessing the value of OSS-based ICT solutions, the Australian Government Information Management Office (AGIMO) has developed the Demand Assessment Methodology (DAM) and Value Assessment Methodology (VAM). Demand and Value were chosen as the two most important criteria in assessing a potential ICT application or service from the perspective of end users and customers.

Demand assessment forces the agency to start with the end user and determine the nature of their needs and how they might be best addressed. Value assessment is a more traditional evaluation, typically centred upon costs and benefits. In the case of government, value considerations also need to take account of social and governance implications as well as intangible benefits.

The DAM and VAM models assist agencies in developing transparent and auditable assessments of demand and value propositions for government online programs. These propositions underpin the business case and assist in substantiating the viability of the initiative. They should also be used to justify resource investment and demonstrate transparency and accountability. The two methodologies provide for a consistent approach across agencies.

For further information, see the AGIMO website:

www.agimo.gov.au/government/damvam

Meeting Australian Government requirements

A Guide to ICT Sourcing provides Australian Government agencies with an overview of the various guidelines, documents and regulations that agencies need to understand when preparing a procurement plan. This section presents a brief overview of these requirements; refer to Appendix B of *A Guide to ICT Sourcing* for further information.

Financial management and accountability

For Australian Government agencies, the *Financial Management and Accountability Act* 1997 (commonly known as the FMA Act) specifies the legal framework for the expenditure of public money. This framework incorporates legislation, regulations and guidelines that set out procedures and requirements for proper accountability and auditing of public expenditure.

For more information, see the Department of Finance and Administration's website:

www.finance.gov.au/finframework/fma_legislation.html

Procurement and best practice guidelines

The Commonwealth Procurement Guidelines (CPG) establishes *value for money* as the guiding principle for Australian Government procurement processes. Agencies are free to define their own procurement practices provided they comply with this guideline and all relevant government administrative requirements covering ICT purchasing.

Agencies must ensure they:

- Comply with the procurement procedures outlined in the CPG;
- Comply with additional requirements established through Chief Executive Instructions (CEIs); and
- Understand their accountability obligations to their Minister, the Government, the Parliament and the public.

In addition, the Procurement Policy Framework provides guidance on:

- Value for money;
- Efficient, effective and ethical use of public resources;
- Accountability and transparency; and
- Industry development requirements.

For more information, see the Department of Finance and Administration website:

www.finance.gov.au/ctc/publications/purchasing/cpg/commonwealth_procurement_guide.html

Chief Executive Instructions (CEIs)

Within an agency, CEIs are the primary source of information and operational guidance for Procurement Officers. These instructions provide an agency-specific financial management framework for procurement.

See *A Guide to ICT Sourcing* for further details about CEIs.

Endorsed Supplier Arrangement

Businesses that want to sell ICT products and services to the Australian Government must achieve pre-qualification through the Endorsed Supplier Arrangement (ESA) scheme. All agencies governed by the FMA Act must only acquire goods and services from endorsed suppliers.

For more information, see the Department of Finance and Administration's ESA website:

www.esa.finance.gov.au

Security management

Security is an important consideration in the procurement of any ICT solution. In general, agencies themselves remain accountable for the security and efficiency of any function that is sourced through an external service provider. The provider is required to meet the levels of security that were established for that function.

For more information, see the Attorney-General's Department website and the Defence Signals Directorate website:

www.ag.gov.au/www/protectivesecurityHome.nsf

www.dsd.gov.au/library/acsi33/acsi33.html

Government Information Technology and Communications (GITC) contracting framework

The Government Information Technology and Communications (GITC) contracting framework is a legal framework established by the Australian Government to provide standard terms and conditions for the purchase of ICT goods and services. The GITC provides models for a head agreement, terms and conditions, contract details and appendices. These allow agencies to construct an appropriate agreement with vendors with greater efficiency.

For more information, see the Department of Finance and Administration's GITC website:

www.gitc.finance.gov.au

Risk analysis and risk management

A core requirement in any change management process is to understand and manage risk. Implementation of an ICT solution is a complex change management process that requires careful planning and execution. This is one of the reasons many ICT projects can fail to meet their original business objectives.

Sourcing OSS solutions is a new and less understood area for Government Agencies. As a result, it often seems to involve higher risk. As open source solutions become more mainstream and agencies gain expertise in evaluating and deploying them, this perception of risk should subside.

The process of deploying solutions based on open source software does not necessarily involve a higher or lower level of risk compared with projects based on proprietary software. There is, however, a change in risk profile. This change needs to be well understood and managed by the agency before undertaking any OSS deployment.

Staff responsible for ICT procurement and project implementation within agencies already have experience in deploying proprietary software technologies. To highlight the differences between these traditional projects and open source software, this section reviews some of the risks involved in both types of project. It is intended to help agencies understand the particular areas that may require additional attention when contemplating open source solutions.

Availability of viable competitors

One high-level risk associated with proprietary software technology (particularly software that is only available from a single publisher or supplier) is the financial risk of potentially high termination costs. This risk arises for a number of reasons, but the most important issue is the lack of an alternative supplier for the software in question.

The result is a lock-in scenario where an agency is tied to a particular supplier with little room for negotiation. This stems from the prohibitively high cost of moving away from a particular piece of technology for which there is no functional or interoperable equivalent from an alternative supplier. Such scenarios allow the current vendor to increase future product pricing, support cost structures or other contractual terms. *A Guide to ICT Sourcing* provides a detailed review of this topic. It suggests that agencies develop a transition/termination strategy during the original procurement process to reduce the risk of future problems for the agency.

While lock-in scenarios are not necessarily eliminated under the open source model, they are easier to avoid. This occurs because of the free redistribution clause of the underlying software licensing. The result is an economic model that encourages competing vendors to enter a particular product space.

No single vendor can obtain a monopoly on the distribution, sale or support of an open source product. Other vendors can adopt a particular software technology for commercial distribution and support if they understand that there is a viable market. As a result, open source products are often available from multiple sources. This provides the possibility of a sideways transition to an alternative vendor that allows the agency to continue using the same open source product. This in turn reduces the costs and risks typically associated with terminating sourcing agreements with proprietary software vendors.

Warranties and indemnities

It is important that purchasers do not overestimate the value of warranties offered with software.

Open source software that is downloaded free generally does not offer warranties for agency users. Purchasers should take care to understand the level of coverage afforded by warranties. Proprietary software warranties normally only offer to use their best efforts to correct defects, or reimburse the software licence fees paid. As OSS products do not require licence fee payments, there is no corresponding offer of reimbursement. Is a matter of judgment whether a particular open source project or a commercial vendor's best efforts are likely to correct defects in a timely manner.

Where an agency acquires open source software through a vendor, the degree of difference may be even smaller. Vendors in such situations are likely to have the same degree of motivation to correct defects as vendors of proprietary software and the supply in both instances is likely to be covered by similar types of contracts with the agency - for example a GITC contract or other appropriate standard form contract. These contracts typically contain a range of warranties and other contractual protections that apply to all the products and services supplied under the contract. Unless there is agreement to the contrary, these would thus extend to open source software supplied by the vendor under the contract.

Bearing these factors in mind, agencies have a choice of sourcing options.

Agencies may decide, after appropriate due diligence and risk assessment, that the absence of indemnities and warranties is a manageable risk in certain scenarios. If this is the case, the agency can opt to procure an OSS solution through an in-house sourcing process as described in this guide.

However, if an agency deems the risks too high, it should only acquire open source solutions through external service providers. Agencies can stipulate appropriate protection for themselves by ensuring that the vendor has executed an appropriate form of supply contract that covers all products and service provided, including open source products.

In either case it is important for agencies to read and understand all software licences and related legal documents. These provide the necessary context for analysing the risks associated with indemnification and warranties.

Lifecycle of open source products

One area where open source and proprietary software differ substantially is in their development, release and usage lifecycles. This results from the differences in development procedures and licensing between the two. Before making any OSS acquisitions, agencies should understand the process by which OSS products reach functional maturity for production-quality deployment.

In general, open source software is released very early in the development cycle and quite often in immature form. These technology previews are intended for technically oriented users or other developers. These groups provide both user feedback and bug fixes to the developer consortium that manages the software product.

Early phase: technology preview

Early-phase OSS products generally have the following attributes:

- A period of rapid change in the technology;
- Testing is available to all, but few have the necessary skills to participate;
- Installation of the software is very complicated;
- New releases may be distributed weekly or more frequently;
- Documentation is minimal;
- There are relatively few users;
- Support is only available via mailing lists; and
- The software is unpolished and lacks many planned features.

In contrast, proprietary software vendors generally do not release software until most of the intended features are in place. They sometimes provide a limited group of testers with 'beta' copies of the software for feedback. When finally released, this software is normally far more polished than corresponding OSS products. Proprietary products are almost always distributed with a complete set of documentation and help files.

Middle phase: early adoption

The middle phase marks a period of slower development, wider adoption by mainstream users and the addition of many new features sought by those mainstream users. Many OSS products never reach this phase. Many fall away during the early phase. Users normally begin production usage of open source software in this phase.

Some of the attributes of OSS during this middle phase are:

- A 'Version 1.0' release is generally robust;
- The technology becomes increasingly polished, with better manuals and easier (possibly automatic) installation and packaging; and
- Commercial vendors offer production-level support for the product.

This phase can last many years, and can lead to robust and trusted systems.

Maintenance phase: long-term support

When an OSS product has accumulated all the core features its users want, it moves into a mode where the only changes made are to resolve bugs or security faults.

The essential attributes of this phase are:

- Stability and maturity in all aspects of the software;
- Products are trusted as a result of many years of solid performance in their roles;
- Most security and functionality faults have been fixed; and
- The user base drives demand for ongoing support and further development.

OSS applications and products that evolve to the maintenance phase often go on to form the foundation of other OSS projects.

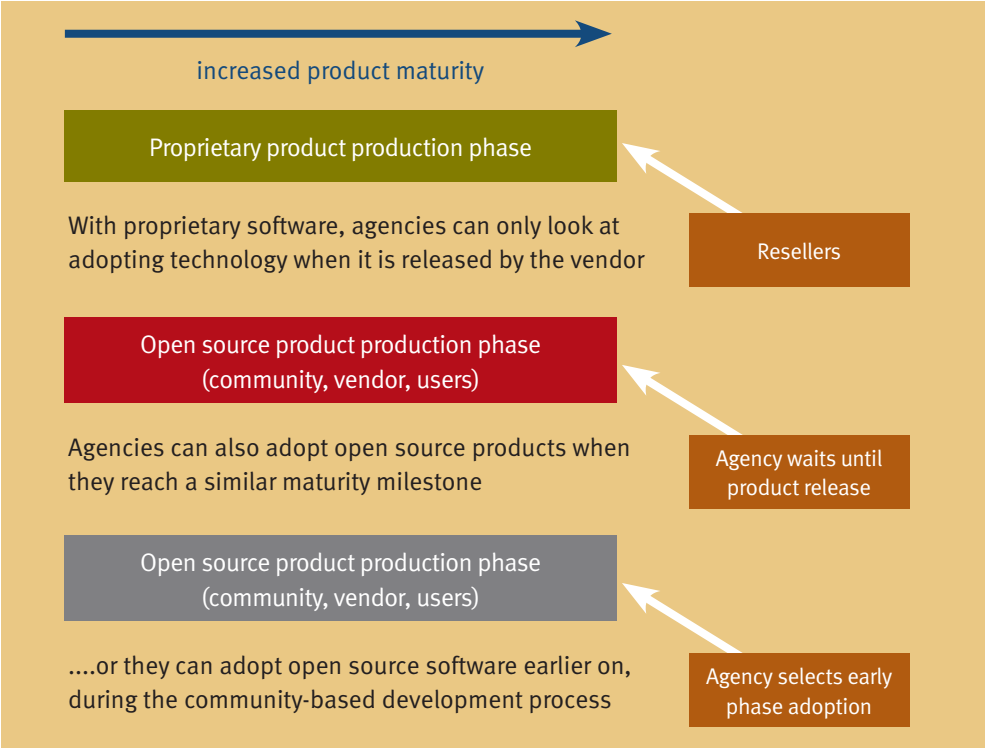
Deciding when to adopt OSS products

Knowing when a particular OSS products is ready for production-grade usage is an important part of any sourcing analysis. Agencies that have appropriate technical expertise and specific requirements may decide to procure an OSS product earlier in its maturation phase than agencies with limited internal capabilities. The advantage for early adopter agencies is that they may have a better chance of ensuring their functionality and interoperability requirements are introduced into the base OSS product at an earlier stage.

However, adopting OSS products too early may lead to reliance on an immature product or software that is inadequately documented. Either of these scenarios brings additional risks that need to be understood and managed. Trialling OSS products in a long-term pilot project is one suggested mechanism. This approach allows an agency to track (and perhaps influence) the development of the product without taking the risk of deploying immature software in a production role.

Figure 6 offers a model flowchart to help agencies determine when they should consider adoption of OSS products.

FIGURE 6. OSS PRODUCT ADOPTION FLOWCHART



Impact of access to source code

Where source code is available, agencies should be aware of the possibility of sourcing products from unauthorised sources. Agencies should only acquire OSS products either directly from a vendor that vouches for the technology or from the primary online repository for a particular product. Many OSS products are also validated with digital signatures to certify that the product was sourced from the originating development group. Agencies should look for such digital signatures to authenticate the product whenever they are downloading the software themselves.

For further detail, see 'What is source code', page 9 and Appendix B, page 52.

Assessing technology risks

The release of a proprietary software product is often accompanied by a significant volume of marketing material such as product data sheets, white papers, brochures and press releases. This material can assist agencies by providing user-friendly information on the functionality and interoperability of the product in question. In addition, industry analysts prepare independent assessments and market intelligence relating to major software releases.

In contrast, OSS software often lacks this technical marketing information. How then can agencies assess that software? How is it possible to gauge fitness for purpose, maturity and interoperability?

This task can be a bit more difficult but need not be impossible. Many popular OSS products have been analysed and positioned in the marketplace by mainstream industry analysts. Analyst reports on these products are available through the usual channels. These reports can give agencies an understanding on the quality and maturity of the software. They also position OSS products within the competitive context of both proprietary and other open source solutions.

CSIRO, the Australian Government research organisation, also conducts analysis in several specific software fields. It produces reports on niche market segments such as the enterprise application server market[3]. Most CSIRO analyses cover open source products and these reports are generally available for agencies.

Sometimes, however, there may be no independent studies on particular open source products of interest to government agencies. In such circumstances, it is still possible for agencies to prepare an informed analysis of the technology risks.

Maturity and fitness for purpose

The first question agenices should ask is whether the software has an established track record. They should also consider how long the product has been available, when it was first released as a production-grade (version 1.0) system and its ongoing development since that release.

If the software has been available for a reasonable time and there appears to be a longstanding community of users, this may support claims that the software performs as advertised.

Take for example an open source database server. In this case, delivering on core functionality would mean:

- Solid performance;
- High levels of interoperability and integration with other system components (application servers, development tools, programming interfaces, etc);
- Extremely high levels of data quality and fidelity;
- High levels of robustness and data availability;
- Solid system recovery facilities; and
- Appropriate data import and export capabilities.

The existence of a sizeable quorum of long-term users generally indicates that the product can be trusted to deliver on such functions. Serious users do not put up with inadequate open source software for long, as there are usually viable alternatives. A low-quality database server quickly gains a poor reputation and users tend to avoid it.

[3] CSIRO Middleware Technology Evaluation Project: www.cmis.csiro.au/adsat/mte.htm.

On the other hand, a product may be old but have relatively few users. This is inconclusive evidence of the value of the technology. A small user base may merely reflect the product's position in a niche market segment. Alternatively, the software may indeed be of poor quality. The 70,000-plus open source packages available are of varying quality.

In cases where the product is solid and meets a market need, open source software can mature rapidly and attain high quality standards in a remarkably short timeframe. However, some project developers and maintainers do a poor job of marketing their product and this can sometimes limit the size of the user base. In such cases, agencies should exercise caution because poor marketing can sometimes cause a project to may falter due to lack of critical mass. Size and momentum are critical success factors for an open source product to ensure it attracts ongoing maintenance and support. These factors are discussed in more detail in the preceding section entitled 'Lifecycle of open source products', page 30.

Analysis of technology roadmap

A technology roadmap provides users (and other stakeholders) of a software product with information about:

- When new features and improvements are planned;
- When new versions will be released; and
- Strategic positioning of the technology, including interoperability, interfaces and support for standards.

By default, the technology roadmap for a particular OSS product is posted on the project's website. Sometimes OSS vendors choose to aggregate multiple open source products; in such cases, the vendor usually provides a roadmap that encompasses the aggregate solution as a whole. Roadmap information provides users with an outline of vendor intentions, status of new versions, strategic direction and end-of-life timelines for commercial support.

Roadmaps produced by open source groups differ from those produced by proprietary software vendors. OSS roadmaps focus more on the technical attributes and functional outlines for the forthcoming versions of the product.

In contrast, product roadmaps from proprietary vendors tend to offer more detail about background business objectives. They also emphasise how a particular product will integrate with that vendor's other products. Proprietary vendors are also more likely to generate roadmaps for overarching platform architecture frameworks and integration methodologies. These are particularly evident if the vendor regards such functions as offering a marketing advantage over rivals.

Agencies are advised to assess architectural frameworks from both open source and proprietary vendors with a critical eye. Many agencies make procurement and strategic decisions based on roadmaps published by vendors, but these often have a limited lifespan and software projects frequently change over time. Agencies should assess a vendor's past history to determine the efficacy of previous roadmaps and the vendor's commitment to delivering on a promised vision. See page 38 for further details on technology roadmaps.

Risk mitigation procedures

Before deciding to adopt any open source product, it is necessary to undertake a careful review of the technology, its history, patterns of use, popularity and feedback from existing users. Depending on the sourcing option, this review is the responsibility of either the agency or the agency in conjunction with its external service provider. The review needs to address the following issues:

- Confirm the software operates as described by the project's website and documentation;
- Ensure performance matches or exceeds the agency's mandatory requirements;
- Confirm the project is being maintained. Indications of this include:
  - ongoing and currently announced and released versions, security fixes and feature updates;
  - updates and announcements published on the project website and information forums;
  - an active user discussion forum or mailing list, archives of which are publicly available;
  - a publicly available project roadmap detailing technology direction; and
  - a growing user base; and
- Confirm the availability of local service providers to support the product (these providers are sometimes listed on the project's website).

Agencies may wish to consider completing this due diligence process for each major component of an open source solution.

A project mailing list or forum is an excellent source of information about the suitability, stability, quality and robustness of a particular technology. Look for comments from users to indicate that they are generally happy. Also look for comments from users indicating that they have been using the software in production for some time with no serious issues.

Assessing sourcing risks

Depending on an agency's sourcing strategy, managing open source procurement may require little or no change to existing business processes. On the other hand, certain scenarios may dictate a considerably different approach. By examining the spectrum of sourcing options, it is possible to highlight any project management changes required.

In-house sourcing

Procurement by direct download of open source software represents a new approach for agenices. As might be expected, downloading any type of software directly from the Internet involves a higher level of risk. Unless agency staff have the particular technical skills required to check the veracity of the download, there are potential issues with infection by malicious software such as viruses, trojans and key-loggers.

Another issue is the question of indemnification and warranties, covered in detail in the previous chapter. Since in-house sourcing of open source solutions does not provide agencies with warranties and indemnities, alternative risk mitigation strategies need to be considered. This may include establishing indemnification insurance policies with a specialised software insurance risk broker. There are insurance firms that specialise in providing such policies to users of open source software.

Agencies need to investigate the risks involved in deploying or migrating to open source solutions. Detailed understanding of the technologies involved increases the likelihood of successful deployment and integration with existing ICT infrastructure and platforms.

Wherever possible, an agency should only adopt open source software if it has identified more than one vendor offering local support. Identifying support vendors in advance is a worthwhile step even when an agency decides to use in-house sourcing. Local support vendors offer a safety net for the agency in situations where in-house expertise is not able to resolve a serious issue.

This is particularly important in situations where the software in question is used for mission-critical activities. Agency technical staff might undertake most day-to-day operational duties. However, the availability of reliable second and third-level technical support is an important step in reducing risk when adopting open source solutions.

If an agency selects an in-house sourcing model, they must ensure technical and procurement staff allocate the time necessary to understand the platforms and components that are expected to form the proposed open source solution. Many solutions are comprised of numerous interlinking components. Familiarity and understanding of the relationships between these components makes it easier to determine if the solution is fit for purpose. This process is covered in detail in Appendix C, “Open source software packaging”, page 58.

External sourcing

The starting point for any external sourcing arrangement is a vendor selection process that includes risk assessment. In this context, deploying OSS solutions is no different to other solutions. The onus is on the provider to cover the agency for any legal risk that may arise. Agencies need to undertake thorough due diligence to mitigate technology risks and change management risks.

If an agency selects an external sourcing model for OSS solutions, the vendor should assume all support responsibilities. Agencies need to ensure that the vendor takes explicit responsibility for all appropriate risk mitigation procedures for the software in question. If the agency has policies about the security standards of all software introduced onto its network, vendors should be required to follow such policies as part of their risk analysis. Compliance with security standards should be included in the commercial agreement, although the agency may need to enact a system of checks to verify compliance.

The agency should require the vendor to undertake a full audit of all software components (both proprietary and open source) to establish their pedigree and verify licence agreements and risk assessment documentation. Software integration issues should be understood and documented. In addition, the agency may undertake its own assessment to verify the vendor’s performance.

Procuring open source solutions through any vendor also introduces risks associated with vendor competence, reach and maturity. These issues may be more acute when dealing with

smaller or less established vendors. Each potential vendor needs to undergo an appropriate due diligence assessment. Such an assessment must cover:

- Financial strength and stability;
- Endorsed Supplier status;
- Risk management credentials;
- Evaluation of internal controls;
- Review of business continuity plan;
- Analysis of third-party exposure;
- Accounting policies and practices;
- General capability overview;
- Commercial management;
- Service delivery and management; and
- Other factors outlined in the Australian Government publication *A Guide to ICT Sourcing*[4].

Assessing long-term product viability

Open source projects succeed and become popular only when they are useful to constituent user communities. To avoid being left with potentially unsupported OSS products, agencies should assess the long-term viability of software options before they enter service. For example, if an open source project has a shrinking user base this could imply a problem with the technology. It could also imply dwindling resources dedicated to the product's upkeep. Either point should be considered as negative when assessing the risk of the product.

While it is possible (if deemed important enough) for a single agency to continue maintaining an open source project after other users have discontinued use, as the last remaining user the agency assumes all maintenance responsibilities for that product. This includes responsibilities for support, security fixes and ongoing technical development. Such a scenario should be avoided unless the software in question is particularly unique and valuable.

Before undertaking such a primary maintenance role, an agency must be able to justify its position. The costs and effort involved in ongoing support of an open source project need to be weighed against the costs and effort involved in moving to more broadly supported software. The costs of migrating to a newer solution may outweigh the advantages, justifying continued maintenance of the existing solution. In such cases, the existing open source codebase can be maintained until this equation changes. Agencies need to factor such scenarios into the agency's risk mitigation strategies.

The aim of this analysis is to reduce the chances of any open source product used in production becoming 'orphaned'. Similar guarantees cannot be made for proprietary products. Once the vendor decides to reduce or terminate support for a particular proprietary technology, agencies often have no alternative source for support, security fixes and ongoing technical development. This risk must also be factored into product acquisition.

Assessing technology roadmap risks

Agencies should seek technology roadmaps for all important software used in production. A technology roadmap is a document that provides an outline of projected future improvements and delivery timelines. Its purpose is to help users of the software to understand what lies ahead.

[4] See page 42 of *A Guide to ICT Sourcing* for further detail.

Obviously there is no guarantee that what a vendor states in its technology roadmap will come to fruition. This is equally true for open source development groups and vendors of proprietary software. However, it is important to understand the future direction or vision of the product whenever possible, as this knowledge helps an agency measure the future obsolescence risks involved in selecting particular technology.

Equally important is the vendor's ability to execute on its vision. A vendor that changes its product roadmap or its general technology platform should be classed as a higher risk provider. The agency has less surety that the vendor's products will evolve the way the agency expects. There may also be less assurance that the product will be supported in years to come.

Agencies should factor in the risks associated with functional variations to the product in future versions. It is possible that the vendor may take the product on paths leading away from the agency's original requirements. Agencies should request vendors to make their future intentions clear in product roadmaps so they can assess strategic ICT impact.

The agency should regularly review the product roadmap – perhaps every six to 12 months – in order to better gauge future risks. Regular reviews can assist the agency to develop an understanding of the vendor's credibility and ability to execute on the roadmap. This review process can then be fed into future planning assumptions.

As a minimum, the roadmap assessment and review process should:

- Locate and record the roadmap presented by the product's developer consortium or vendor;
- Locate and record information that the vendor offers to its client base and user community, including information about the latest developments, product release dates and versions;
- Check how current the timeline is. For example, if the timeline references future events that are now months or years in the past, this can indicate an information repository that is no longer well maintained – which may in turn signal problems with the development consortium or vendor;
- Contact the nominated representative of the consortium or vendor. Seek information on the frequency of updates to the roadmap and release schedules;
- Once this timetable is established, locate and record separate copies of web pages containing both the product roadmap and release schedules;
- Compare the latest roadmap with previous roadmaps, noting modifications. Specifically, look for the removal of milestones that were actually achieved and the addition of future product milestones;
- Compare the software developer's latest news items web page with previous versions, noting any additions;
- Compare the timeline and dates found on the current news items web page to confirm how many of the originally proposed milestones were achieved. Note this number for future reference; and
- Repeat this cycle for as long as your analysis period requires.

In time, agencies should be able to determine with some precision how well different open source developer consortia and vendors keep to their published timelines. This information should be tabled as part of the risk assessment for that particular product or vendor.

This process can also alert an agency to problems that may arise in the core group that produces and maintains a particular piece of software. Early warning of problems can lead to more effective management of the risks involved in transitioning from such software.

Managing risks in technology roadmap variation

Some open source development groups or vendors might not provide a technology roadmap of sufficient quality and detail. Alternatively, they may not consistently adhere to their own

roadmaps. Such projects may make it harder for agencies to plan future strategies. This is another risk that needs to be managed.

However, in some scenarios the lack of a consistently executed roadmap may be less of a concern. For example, if an agency selects or uses a mature and widely used OSS product, the risks involved in using such software are generally low. The software normally has all the features it needs to fulfil the needs of the agency and other users. New features are rarely necessary.

Similarly, if the product forms part of fixed system infrastructure that works well and the agency has no future plans or requirements for enhancement, then the lack of a roadmap poses minimal risk. Many open source infrastructure components change slowly once they mature. As a result, the development group may not feel a need for lengthy discussion of future directions, which may be the reason for a lack of technology roadmap.

The following checklist may be useful for agencies when considering procurement of open source software and services.

Due diligence risk mitigation checklist

Attribute	Yes/No
Does the software operate as described by the project's website and documentation?	
Does this operation match or exceed mandatory requirements for procurement?	
Is the software supported by a publicly accessible website that lists recent announcements?	
Does the software's website provide information on new versions?	
Does the software's website provide a running history of previous versions so you can check the frequency of releases?	
Does the software's website provide information on security fixes and feature updates?	
Does the software have a publicly accessible and independent user forum or mailing list?	
Does this forum have archives that can be searched either in-site or through a public search engine?	
Does the software have a published roadmap available on the project website?	
Does this roadmap meet the agency's requirements?	
If the project needs additional features to meet agency requirements, is there an obvious process for requesting new features or enhancements from the development team?	
Is there a general growth trend in the number of users?	
Are there local service providers listed on the project website?	
Have you performed a risk assessment of these service providers?	
Have you analysed the requirements of any ancillary components, modules, libraries or systems the software needs to operate properly?	
Have you prepared a risk mitigation checklist for each of these sub-components?	

Understanding the legal context

The growing popularity of open source software has encouraged greater interest in software licensing. This interest is intensified by the fact that open source licences are very different to traditional proprietary software licences.

Most proprietary licences contain similar terms and users of such software have accumulated a broad understanding of these licence terms. Proprietary software licences focus on what the user may or may not do with the software in question (particularly relating to the number of systems on which the software can be installed or the number of concurrent users).

In contrast, open source licences are generally not concerned with how users use the software at all. The focus is on redistribution of the software and continued access to source code.

As a result, proprietary software has a different risk profile to open source software. Agencies need to understand and manage these risks.

Assessing licence risks

Most open source software is released under one of a number of different licensing schemes. There are several dozen such licensing schemes in common usage. This high number reflects the fact that most open source software is developed by consortia of developers or by vendors and each has different expectations about how their software should be reused by downstream developers. In some cases, developers who find that an existing licence does not fulfil their requirements may simply formulate a new one.

Commentators often complain that there are too many open source licences and that this adds to the confusion and therefore risk for users. While many in the open source industry would prefer to restrict further fragmentation in licensing, there is no mechanism to prevent new licences being added to the list maintained by the Open Source Initiative[5].

Not all software categorised as open source has an explicit licence governing its use and redistribution. For example, public domain software (software with no copyright attached) has no restrictions or requirements pertaining to subsequent use and redistribution.

Licence auditing should be considered best practice for both OSS and proprietary products. To mitigate risk, agencies should understand every product licence used within their environments, both proprietary and open source.

[5] OSI list of open source licences: www.opensource.org/licenses/index.php.

OSS licence types

All licences certified by the Open Source Initiative (OSI) as compliant with the Open Source Definition[6] share important characteristics. The most important of these are specific freedoms for users who run the software. However, even OSI-compliant licences have important differences, particularly in the redistribution of source and binary code.

OSI licences generally fall into two distinct categories:

- Attribution-style licences, exemplified by the Berkeley Software Distribution (BSD) licence; and
- Share and share alike licences, exemplified by the GNU General Public Licence (GPL).

Most OSI licences have minimal variation in terms of usage risk (as opposed to redistribution risk).

Attribution licences allow other developers to take some or all of the source code from a codebase and reuse it within another software codebase. The copyright and licence references must be kept intact and attribution statements must be added to the derived software. The derivative software can be re-released under any other licence, including closed source proprietary licences. This facility results in situations where attribution-licensed software can be (and has been) embedded within many proprietary products already used by agencies.

Software licensed under a share-and-share-alike scheme requires that any modifications, derivations or redistributions of the original software that are made available to a third party must then be made openly available. It is therefore not possible to take such software and make it proprietary. Developers of software based on share-and-share-alike schemes demand that their software remain open and free in perpetuity.

Implications of open source copyright

The most common open source licence is the GNU General Public Licence (GPL). This licence ensures that downstream programmers adhere to the original programmer's licensing requirement for free redistribution of the source code. Like most open source software, GPL-licensed software is protected by copyright.

The GPL uses copyright protection in a manner that ensures the continued availability of the underlying product source code. Agencies that intend to modify GPL-licensed products and redistribute those modifications should understand the legal implications of such a move. The legal intricacies of redistributing open source software are discussed in 'Sharing OSS solutions', page 47. Agencies that merely use an OSS package and do not intend to redistribute the software have no additional risk management issues in this area.

The GPL legally enforces its mandate by allowing free redistribution of the software only when the licence is accepted as a whole. Failure to comply with the full licence removes the right of the downstream developer to copy the GPL-licensed code.

Downstream programmers who want to modify a GPL-licensed codebase and redistribute the resulting software as a product must comply with the GPL conditions in full. The licence specifies that modifications to source code, if released to a third party, must be made available to downstream users. A modifier must also allow users of their product to modify the source code under GPL terms or their right to redistribute the software is revoked.

Contrasting open source and proprietary licences

There are almost as many proprietary licences as there are proprietary software packages. Proprietary software licences generally focus on the use of the software. This may be defined in terms of the size, scale or types of use that are permitted. Most proprietary licences:

[6] OSI definition of open source: www.opensource.org/docs/definition.php.

- Restrict the user in the number of systems (PCs, servers) on which the software can be installed;
- Limit the number of processors those systems can have;
- Limit the number of simultaneous users: the more users running the software, the more licences an agency needs to purchase;
- Prohibit users from copying the software (beyond backup copies);
- Prohibit decompilation or interoperability engineering (reverse engineering), including activities that are necessary to create software that can transfer data to and from an application; and
- Include additional esoteric requirements such as stipulations that the user must not use the software to create and publish material disparaging the vendor. Other licences limit the user's ability to publish information or the results of benchmarking tests that compare the product to competing products.

In contrast, open source software licences are not concerned with the mode or scale of usage. Most of these limitations have no comparable requirements in OSS licensing schemes. All open source licences involve similar risks for users. Where there is variation between licences, this usually relates to the way source code can be extended, enhanced, embedded or redistributed.

Contrasting different open source licences

It is important for agencies to understand software licences before agreeing to use a product, be it proprietary or open source. When using an OSS product, an agency should also consider the impact of the software licence if it is contemplating modifying the OSS package and redistributing the resulting software to other agencies or the wider ICT community. If an agency is planning source code modifications, then a deeper understanding of the nuances of the software's licence is mandatory. See the chapter on 'Sharing OSS solutions', page 47, for detailed discussion of this issue.

Modification scenarios

There are several scenarios to consider that may involve modification of an OSS package:

- **Scenario 1:** *An agency takes an existing open source product and extends it to fit the agency's specific purposes and requirements. The agency plans only to use the resulting software in-house.*

  Applicable OSS licences: All open source software licences allow for this scenario. The agency can proceed regardless of what open source licence the product was originally released under.

- **Scenario 2:** *In a slight modification of the previous example, an agency takes an existing open source product and extends it to fit specific agency purposes and requirements. The agency then wishes to redistribute the resulting technology to other Australian Government agencies.*

  Applicable OSS licences: As the Commonwealth is a single legal entity (despite the fact that it is represented by many Australian Government agencies), making modified versions of open source software available to others within those agencies does not amount to redistribution that may trigger obligations under the terms of the relevant open source licence. However distribution to Australian Government entities that are independent legal entities would trigger such obligations.

- **Scenario 3:** *An agency takes an existing open source product and extends it to fit the agency's specific purposes and requirements. The agency wants to allow distribution of the modified software beyond the Australian Government.*

  Applicable OSS licences: All open source software licences allow for this scenario. However, some OSS licences require the agency to provide the modified source code to the recipients of the modified programs.

- **Scenario 4:** *An agency takes an existing open source product and extends it to fit the agency's specific purposes and requirements. The agency then wishes to distribute a 'closed source' version of the software.*

  Applicable OSS licences: Some OSS licences, such as the General Public Licence (GPL), impose restrictions that prevent this activity. To enable redistribution of a modified OSS product in closed source format, agencies may wish to consider open source products with licences that permit the process of 'closing off' the open source software codebase. The Berkeley Software Distribution (BSD) licence is one example of a licence that permits closing off.

- **Scenario 5:** *An agency wishes to redistribute a closed source (proprietary) product that it has developed. It also plans to package the software alongside open source modules, components, libraries or applications.*

  Applicable OSS licences: If the proprietary product does not directly link into these other technologies, the agency is free to redistribute this bundling. Nevertheless it is prudent to read and understand the licence terms of each bundled technology to reduce legal risks.

- **Scenario 6:** *An agency plans to redistribute a software product (either open source or closed source) it has developed. The software has direct links into open source modules, components or libraries.*

  Applicable OSS licences: The agency must understand and comply with the licence of each product its software links into. If any of these products are licensed under the GPL scheme, then source code must be provided to users of the resulting product. If the linked products are licensed under the Lesser General Public Licence (LGPL) scheme (which allows users to create closed source modifications without having to provide open source code) or an attribution licence such as BSD, then source code need not be provided.

SOFTWARE LICENSING, WHETHER OPEN SOURCE OR PROPRIETARY, IS A COMPLEX LEGAL AREA. APPROPRIATE LEGAL ADVICE SHOULD BE OBTAINED BEFORE ENTERING INTO A LICENSING AGREEMENT.

Figure 7 provides a matrix showing which open source licences are available for each type of in-house development and redistribution situation.

FIGURE 7. DECISION MATRIX FOR OPEN SOURCE LICENCES[7]

Action to be taken with open source software	Licences available	Specific licences you can use
Do not plan to modify source code	All open source licences	GPL, BSD, Mozilla Public Licence, MIT Licence, LGPL
Plan to modify source code	All open source licences	GPL, BSD, Mozilla Public Licence, MIT Licence, LGPL
Plan to modify source code and distribute only within the Australian Government	All open source licences	GPL, BSD, Mozilla Public Licence, MIT Licence, LGPL
Plan to modify source code and distribute beyond the Australian Government as an open source product	All open source licences	GPL, BSD, Mozilla Public Licence, MIT Licence, LGPL
Plan to modify source code and distribute beyond the Australian Government as a proprietary product	Cannot use open source licences that contain a "share and share alike" clause mandating open release of modified code in perpetuity	BSD, MIT Licence
Plan to link open source product with internally developed code and distribute beyond the Australian Government as a proprietary product	Cannot use open source licences that contain a "share and share alike" clause mandating open release of modified code in perpetuity	BSD, MIT Licence, LGPL

Examining open source licences in detail

As a matter of best practice, agencies are advised to ensure they understand the licences that apply to all software they wish to use, including both proprietary and open source solutions. This reduces the risk that an agency may inadvertently contravene a particular licence condition. This requirement goes beyond merely ensuring that each user is licensed to run a copy of that software.

As discussed above, open source software licences are not generally concerned with what users do with the software. They usually focus on how other downstream software developers and software publishers may modify or redistribute the software.

Figure 7 (above) provides a quick reference matrix to help agencies understand the main variations among open source licences. This table is not a substitute for a close examination of the actual software licence provided with a product.

[7] Note that this matrix does not cover all OSS licences, only the most commonly used licences. For a full list of OSS licences, see the Open Source Initiative website: www.opensource.org.

Applicability of open source licences in Australia[8]

As explained above, Australian Government agencies procure and use software under many different types of licences. For effective risk analysis, it is important to understand these licences and their application in the context of the Australian legal framework.

Few software licences (proprietary or open source) have been fully tested through the legal system. Indeed, there have been no known instances of an open source licence being defended or tested in an Australian court. Internationally, only a small number of cases have arisen where the most popular open source licence, the General Public Licence, was tested in court.

Procurement of OSS solutions involves a number of business issues that are not completely addressed by open source licences. For example, there is no definition of the "ancillary services" that a vendor is to provide as part of the solution, including:

- Assessment of customer needs;
- Installation and integration services;
- Training; and
- Ongoing support services.

Many of these issues would normally be covered by procurement contracts such as the vendor services agreement or a support contract, rather than the software licence. This approach can be applied to either proprietary or OSS solutions.

In addition, other matters of interest are ill-defined, including stipulation of the governing jurisdiction, alternative dispute resolution procedures, confidentiality issues, taxation issues and so on. Agencies should ensure such ancillary considerations are addressed in the contractual framework through it undertakes procurement.

Trade Practices Act considerations

In some circumstances, OSS licences appear to exclude all warranties about product quality. However, agencies and vendors need to consider the impact of Trade Practices Act 1974 (TPA). The TPA imposes certain implied warranties in all sales contracts and many of these warranties cannot be excluded by contract. The TPA is complemented by various state and territory laws covering fair trading and the sale of goods.

Agencies need to take particular care in situations when they may redistribute or sell an OSS-based solution to other organisations outside the sphere of the Australian Government. In such cases, it is prudent to obtain legal advice about the impact of TPA and sales of goods laws.

Broader intellectual property implications

Intellectual property (IP) issues are often confusing for people without a legal background. Nevertheless, it is important for agencies to understand how different strands of intellectual property law relate to software. Many of these issues are particularly important in the open source industry. This section provides an overview of the topic, specifically highlighting intellectual property issues with implications for assessing risks related to OSS procurement. The Department of Communications, Information Technology and the Arts (DCITA) has produced material covering these issues in greater detail[9].

[8] Some of the ideas for this section are based on arguments presented by Peter C J James, a partner at the law firm Allens Arthur Robinson, during the 'Legal Issues Relating to Free and Open Source Software' conference at Queensland University of Technology School of Law in 2003. The paper has been published in the conference proceedings: www.law.qut.edu.au/files/open_source_book.pdf.

[9] Commonwealth IT IP Guidelines: www.dcita.gov.au/ip/commonwealth_it_ip_guidelines.

Agencies face a different and possibly less defined intellectual property environment in the OSS space owing to the open nature of the source code and the different types of licensing used for OSS products. Some users who are new to open source assume that there are no intellectual property boundaries and that everyone is free to reuse any intellectual property contained in OSS products. This is not the case.

If anything, open access to the underlying intellectual property means IP subtleties are perhaps even more important for open source software. Agency staff should be well briefed in these subtleties to ensure they have the analytical and decision-making skills required to judge and manage the different risks involved.

Who owns and controls open source software

Although the authors of some software may have waived intellectual property rights to open software they have created (i.e. it has been donated to the 'public domain'), the copyright in most open source software is owned by the respective developers of the code in exactly the same way as proprietary software.

Just as the owners of proprietary software prevent unauthorised use of their code by licensing their software, open source software owners do the same thing. It is only the terms of the licence that are different. Rather than restricting use, an open source licence seeks to prevent licensees from restricting the further development and redistribution of the code.

Copyright and legal enforceability of licence agreements are thus just as fundamental to open source software as they are to proprietary software.

It is usually the case that an open source software product has many contributing authors and thus potentially many different copyright owners. To date this has not proved to be an impediment to effective enforcement of open source licences, but some open source projects have taken steps to simplify the ownership arrangements by arranging for all contributors to assign their copyright to an appropriate legal entity that becomes the owner/guardian of the code. This single entity can then take enforcement action if it becomes necessary in much the same way as a proprietary software company would take action to protect its software.

If the author of open source software retains ownership of the code, they can continue to exercise the benefits of ownership concurrently with the rights they have irrevocably granted to open source users. For example, an author can release both open source and proprietary licensed versions. What the open source concept is designed to ensure is that the author cannot use legal means to control the future of the code within the open source licence scope once it is released.

So, different types of software are subject to different risks. Although open source software's development effort is subject to the skills and enthusiasm of its supporting community, it is not subject to other risks that are part and parcel of the proprietary software model, most notably that a product can whither and die when the copyright owner is taken over by another company.

Similarly, although some aspects of open source software development may make them seemingly more vulnerable to intellectual property infringement claims than proprietary software, well managed projects are subject to close peer scrutiny which tends to mitigate against such risks. The reality is that both proprietary and open source could be subject to infringement claims and that agencies should always think through what options they have if their ability to use a product is disrupted or they find themselves embroiled in infringement litigation as a user of a software product.

The value of indemnities from a vendor should not be overlooked, though their scope needs to be legally scrutinised and the financial ability of the vendor to stand behind them assessed, should a major problem occur. Insurance policies to indemnify users of major open source products are also beginning to emerge and might also be considered as a risk management tool.

Sharing OSS solutions

In most situations, open source solutions procured by an agency do not require tailored code modification, so deployment can proceed in a straightforward manner. However, open source solutions offer broad scope for customisation by either the originating vendor, the agency or a third party. In such situations, agencies need to consider several possible scenarios.

Modifying products for use within one agency

In one scenario, an agency requires source code modifications but does not expect the resulting software to be used by other agencies or the broader ICT industry. During contract negotiations, the agency and the vendor must resolve several issues (preferably in explicit contractual terms) including:

- Whether any code modification is required;
- Commercial fees and scope for such modification; and
- Whether the resulting work is to become the copyright property of the agency (and therefore the Australian Government).

The agency should ensure the commercial agreement stipulates that it receives copies (in machine-readable electronic form) of such code modifications. As part of a standard risk mitigation process, it is good practice for an agency to maintain multiple copies in a safe and secure place. These code modifications can be reapplied to an open source product to recreate the original solution, if necessary.

The agency is not required to re-release any source code developed during the project into the open source community. The code enhancement is the property of the Australian Government, and the agency may decide that the code is only suitable for use within the Australian Government.

The only widely used open source licence that mandates re-release of source code enhancements is the GNU General Public Licence (GPL). However, the GPL does not automatically require a user to release any modifications it makes to source code; it only stipulates that when an agency chooses to release such modifications it does so under GPL conditions.

Specifically, GPL requires downstream developers (in this case, the OSS solution vendor) to provide the source code for any enhancement to their customer (the agency). In other words, the OSS solution vendor has an onus to ensure that the client agency receives the full source code to any modifications; there is no requirement for the solution vendor to provide such source code to anyone else. At no point does the GPL mandate that the vendor (or the agency client) must release copies of the modified source code to anyone else. If the agency wants to keep that code internal to the agency, it is legally entitled to do so.

An agency may decide, after risk assessment and due diligence, that there are no issues with releasing the code. It may also be possible to negotiate a reduced services fee if the modifications developed for the agency are re-assigned back to the vendor for subsequent release into the open source community.

Some agencies could consider such a strategy as a potential way to reduce the cost of the OSS-based solution. Many smaller open source vendors are interested in extending the capabilities of their products using this process. Agencies should only agree to such a process after assessing and documenting the necessary security and privacy implications in advance.

In general, such cases allow agencies to follow standard project management procedures associated with externally sourced software development activity. The principal difference centres on the possibility of code enhancements being made available to others; this is discussed in detail in the next section.

Sharing modified products with other Australian Government agencies

In this scenario, one agency procures an open source solution and requests modifications. It then wants to make the enhanced solution available to other Australian Government agencies. Assuming the original product codebase is open source, there are two ways to achieve this outcome.

The first is through direct technology transfer between agencies. As per the previous section, any add-on code created when enhancing the software should also be owned by the Australian Government. As a result, other Australian Government Agencies may simply decide to adopt the product through an in-house sourcing process. They are legally able to do so without licensing restrictions, as the Crown is considered a single legal entity.

Alternatively, the downstream agency can negotiate with the original vendor for the vendor to supply a solution based on the enhanced product. In such cases, the agency should be able to negotiate an attractive price because the cost of developing the enhanced product was already borne by the first agency. Further cost efficiencies may result from establishing a multi-agency procurement framework as part of the original agreement. This approach can give the Australian Government additional volume purchasing leverage with the vendor.

A recent example of this approach in action was the release of an open source content management system (CMS) developed for the Australian Government Information Management Office (AGIMO) by SME technology company Squiz.net using MySource Matrix, an open source CMS platform. Through a practice known as 'white-branding', AGIMO and the vendor established a robust and flexible software platform that is now available to other agencies at no upfront cost. For details, see the MySource Matrix White-Branding Documentation Suite published by AGIMO – a business group within the Department of Finance and Administration.

Unless an agency has agreed to permit the general re-release of code enhancements to open source products, agreements with the vendor should stipulate that the modified code remains the property of the Australian Government. An agency must manage such a requirement as part of its overall governance of the commercial agreement.

Making modified products available to the open source community

Agencies that decide from the outset to make any project-scope enhancements available to the wider open source community enjoy a number of benefits. The most important advantage is to avoid what programmers call a 'code fork'.

Code forks occur when the codebase of a product is modified and these modifications are not applied back to the main product code repository. This marks the beginning of divergent processes for development, bug fixes and security fixes. Over time, these divergent codebases will continue to grow apart unless some effort is made to rejoin them. In the worst case, the two codebases can evolve into incompatible technologies. Such a scenario has risk and cost implications for agencies that use the software.

For example, if an agency requests enhancements to a product but does not allow the vendor to apply those changes back to the main code repository, the agency will be running a version of the product that may be increasingly different to the versions used by other clients. Over time, it may be difficult to re-integrate the two diverging codebases. This situation introduces several risks for an agency, including the risk of source code instability and immaturity. Because this code forking process represents the beginning of a different product, the agency must bear the brunt of all future testing and trouble-shooting for its unique environment.

For the vendor it is almost twice as expensive to maintain, enhance, test and fix two divergent codebases. Hence, the vendor may reduce its focus on quality or increase prices to accommodate the requirement to maintain two divergent codebases. Furthermore, an agency can be disadvantaged by not having access to the latest bug fixes, security updates and performance enhancements available to users of the mainstream OSS product. Such possibilities must be factored into the agency's risk assessment, mitigation planning and contract governance.

An agency can alleviate such a dilemma in one of two ways. It can ensure that the code enhancements are made in a modular manner. This allows the vendor to deploy and maintain the standard product codebase for the agency's requirements and add/maintain the enhancements separately.

Alternatively, the agency can authorise the vendor to incorporate its enhancements back into the main codebase[10]. There are several benefits to this approach.

Firstly, it expands the user base for the software, with each additional user providing an ongoing real-world testing environment for the enhancements. Flaws and security issues are likely to be located at a faster rate. This gives the agency stronger and more mature code sooner.

Secondly, it facilitates the process of developing, testing and deploying subsequent versions of the software. If the agency decides to separate its enhancements into a unique version for the agency, it faces substantial re-integration costs and increased migration and operational risks in every product upgrade cycle.

Enhanced code can be released back to the OSS community even if the agency decides to retain copyright over the code enhancements, although this may expose the Australian Government to claims from downstream users of the software.

However, it is possible to assign copyright for the enhancement to the vendor. This should only be done on the proviso that such code will be re-released under an open source licence. Such a precaution should be stipulated in the agreement with the vendor. This ensures that the Australian Government has open access to the product in future, avoiding the risk of paying again for the software. Agencies should seek legal advice to establish a framework whereby responsibility for warranty and indemnity issues rests with the vendor.

[10] Obviously this only makes sense if the enhanced code passes the agency's security and privacy requirements. In addition, the enhanced code needs to useful in other contexts. If the enhancements are merely applicable to the agency itself, the vendor will not normally be interested in offering the code for the mainstream product. Sometimes proper consideration and system design can provide the needed levels of generalisation.

Appendix A: Open source software resources

This section lists online repositories that offer free download of open source software products. It also provides links to other resources providing further information about OSS solutions and the OSS industry. This is not intended to be a comprehensive list: there are many other resources available to agencies and the list continues to grow.

Site name	Web address
AGIMO Open Source Software	www.agimo.gov.au/infrastructure/oss
Apache Software Foundation	www.apache.org
Australian Unix and Open Systems Users Group (AUUG)	www.auug.org.au
Dravis Group report: Open Source Software: Perspectives for Development	www.infodev.org/symp2003/publications/OpenSourceSoftware.pdf
European Commission Free and Open Source Software project	europa.eu.int/information_society/activities/opensource/index_en.htm
Freshmeat	www.freshmeat.net
GNU/Free Software Foundation	www.gnu.org
GNUWin II	gnuwin.epfl.ch/en/index.html
Government Open Code Collaborative (GOCC)	www.gocc.gov
Java-Source.net	www.java-source.net
Linux Australia	www.linux.org.au
Linux HQ	www.linuxhq.com
Linux Software Equivalents	linuxshop.ru/linuxbegin/win-lin-soft-en/table.shtml
LinuxWorld	www.linuxworld.com
Mac OS X Open Source Directory	www.apple.com/downloads/macosx/unix_open_source/
Open Source Industry Australia	www.osia.net.au
Open Software Initiative	www.opensource.org
Open Source Software Educational Society	www.softpanorama.org
Open Source Software Institute (OSSI)	www.oss-institute.org
OSdir.com	www.osdir.com
Samba	www.samba.org
Scientific Applications on Linux	ftp.llp.fu-berlin.de/lsoft/index.shtml
Slashdot	www.slashdot.org
SourceForge	www.sourceforge.net
Tigris.org	www.tigris.org
World Wide Web Consortium	www.w3.org/Status

Description
Introductory information from the Australian Government Information Management Office
Apache provides a suite of open source software projects, including the world's most widely used web server platform
Australia's peak industry body for Unix, Linux and open source professionals
Report on public sector usage of OSS technology prepared for the World Bank's Infodev group
A European Commission site covering free and open source software
A large index of open source software arranged into categories
Home page for the GNU operating system, Free Software Foundation and the GNU General Public Library
A collection of ready-to-run open source software for Windows computers
A collaboration between public sector entities and non-profit academic institutions to encourage sharing of computer code developed for and by government agencies
A site for OSS developers using the Java programming language
Coordinating body for the Australian Linux community
A reliable resource for Linux products
A resource identifying possible Linux equivalents to common Windows-based proprietary software packages
An online news and information source for the Linux operating system
Open source software for Apple's Unix-based Mac OS X operating system
One-stop reference site for the local OSS industry
Non-profit body dedicated to managing and promoting the Open Source Definition
An 'anti-hype' site that provides a critical perspective on OSS
OSSI is a non-profit organisation formed to promote open source software solutions within US federal and state government agencies and academic entities
A directory focused on stable open source applications
An Australian-based open source software initiative providing interoperability between computers running Linux/Unix and Windows
A collection of information and links about Linux-based software for scientists and engineers
An online news and discussion site with an emphasis on OSS issues
A large repository of open source software and development tools
A mid-sized open source community focused on building better tools for collaborative software development
All W3C software is open source/free software compatible with the GPL

Appendix B:
More about open source software

Open source software or free software?

The name 'open source' as a label for a particular class of software has only existed in common usage for a few years. Before this time, such software was generally labelled 'free software'.

In generic terms, free software denotes software that can be acquired without financial remittance. However the technologists who coined the term free software had a very specific intention in mind. The purveyors of free software, led by an organisation called the Free Software Foundation[11], outline the following attributes for free software programs:

- Freedom to run the program for any purpose;
- Freedom to study how the program works and adapt it to a user's own needs, thus the source must be provided;
- Freedom to redistribute copies of the software; and
- Freedom to improve the program and release improvements to the public, so that the whole community benefits from the distribution of source code.

Source: www.fsf.org/philosophy/free-sw.html

The term 'open source' was introduced because the term free software was seen as overlapping too broadly with the freeware/shareware class of software. While freeware and shareware are free to download, free to redistribute and sometimes free to use, they are not supplied with the source code nor the freedom to modify or improve such source code.

Open source business models

One common question about open source software relates to the rationale for its development and publication. Some people do not see why the individuals and vendors who choose to build and maintain complex software do not charge licence fees.

This question is particularly relevant in situations where an agency is trying to determine the long-term viability of a product. Open source software that is devised for short-term reasons and without a sustainable business or community-participation model may not continue to be maintained. Agencies should seek to identify any such software and include this as a part of their risk analysis.

[11] Free Software Foundation: www.fsf.org.

There is no single response to the question of why people contribute to OSS projects. Reasons for contributions to OSS projects may include:

- Deriving non-financial rewards from writing highly functional and well-crafted software;
- Making money from helping customers implement the software;
- Releasing open source software as part of a larger strategic decision to benefit from the OSS distribution model and the growing market for such software. This may sacrifice potential licence fees for the opportunity of gaining revenue through support contracts, consulting services or training;
- Making products that were previously proprietary available as open source. In such cases, the vendor may consider this move to be good corporate responsibility, to ensure its customers are not left with orphaned technology;
- Reducing the market space available for a competitor's product; and
- Leveraging the competitive research and development costs of open source software to lower the cost of a bundled solution;

A number of open source software vendors release products under both an open source licence and a proprietary licence. This approach can work well in situations such as sub-components or libraries. Releasing software under an appropriate open source licence increases the likelihood of broad adoption without expensive marketing. In such situations, other developers may want to build proprietary technology using the library. They generally need to opt for the version licensed on proprietary terms. Such terms generally include a royalty component, which is part of the original vendor's revenue model.

Source code access: implications for agencies

One of the most marked differences between OSS and proprietary software is that the end user has access to the programming source code that makes the software function. With proprietary software, this source code is normally not available to the user.

In many circumstances, OSS licences (such as the GNU General Public Library) require software developers or product vendors to furnish the client with information about exactly where they can find and retrieve the software source code for any OSS products they supply. Usually this source code is available from an online software repository or sometimes on physical installation media (CD-ROM, DVD-ROM).

For most end users access to source code has little practical impact. End users running an OSS package utilise the software in its machine-readable binary code form, rather than in its human-readable source code form. This situation is exactly the same for both OSS and proprietary software tools.

Nevertheless, access to source code provides a number of benefits for any potential user of OSS. For example, open access to source code means it is usually possible to acquire the same OSS solution from more than one independent software vendor (ISV). It is also possible to receive implementation and maintenance support (including programmatic problem resolution services such as bug fixes, security fixes and service packs) from more than one supplier. Some suppliers may even support competitive product lines.

These attributes are particularly valuable during end-of-contract negotiations with suppliers. They typically offer government agencies additional flexibility in such negotiatons. For more information on termination strategy, see 'Phase II: Decide Sourcing Strategy' in *A Guide to ICT Sourcing*.

Assured access to source code also minimises the likelihood that an agency may end up using obsolete or discontinued software, as the source code allows another vendor or

even the agency itself to continue maintaining a software platform. In addition, source code access reduces the potential for vendor lock-in, where an agency has no choice but to continue procuring its ICT solutions through a particular supplier. Such situations significantly reduce the agency's bargaining power and can lead to higher ICT costs.

Open source, open standards and open systems

Three common ICT terms – 'open source', 'open standards' and 'open systems' – are often used interchangeably. However, each term represents quite a different concept, although on some occasions they do overlap. It is important for agencies to differentiate between these ideas when sourcing ICT solutions.

Open source

Open source software is a type of computer software defined by several specific attributes that relate to its licensing and legal framework. Often it also involves a distinctive development and distribution model.

At present, the primary arbiter of what constitutes open source software is the Open Source Initiative[12]. This Initiative sets out various rights and obligations for developers, distributors and users of OSS. These rules define the basic licence conditions under which software must be released to be considered open source. These licence conditions give the users of OSS the right to:

- Use the software for any purpose;
- Make copies of the software for any purpose;
- Access or modify the source code of the software for any purpose; and
- Without payment of a royalty or other fee, distribute copies of:
  - the software (including distributing the software as part of an aggregate distribution containing software from several different sources);
  - a derived or modified form of the software (either in compiled form or as source code), under the same terms as the licence applying to the software.

Source: www.opensource.org/documents/definition.php

Open standards

An open standard is a detailed, descriptive overview of a process, protocol or format. It is formulated through stakeholder consensus. It must be openly published and there should also be no legal or intellectual property restrictions.

Open standards are generally defined by focus groups within standards organisations. The most important standards bodies within the ICT industry include:

- American National Standards Institute (ANSI);
- European Computer Manufacturers Association (ECMA);
- Institute of Electrical and Electronic Engineers (IEEE);
- Internet Engineering Taskforce (IETF);
- International Standards Organisation (ISO);
- Open Group; and
- World Wide Web Consortium (W3C).

[12] Open Source Initiative: www.opensource.org.

In addition, many specific fields within ICT have their own industry standards.

Open standards are not unique to the ICT industry. Many standards are defined and certified by bodies such as Standards Australia. Standards benefit industries because they encourage market growth by allaying customer fears about interoperability. They also ensure that competitive economic pressures exist within given markets, producing price benefits for consumers.

Open standards are increasingly important for ensuring interoperability among competing vendors. It is therefore good practice for agencies to analyse and understand which open standards may be applicable to specific ICT solutions. Products that comply with open standards are easier to integrate into environments designed to use the same standards. Open standards also make it easier for an agency to terminate its relationship with one vendor (or product) and adopt another, hence reducing the risk of vendor lock-in or product lock-in.

Establishing a vendor or product's compliance with open standards often forms part of an agency's risk assessment and due diligence process. It could also form part of the decision matrix used when identifying prime ICT sourcing options, whether open source or not.

Open source software is largely – but not necessarily – based on applicable open standards. However, agencies or their ICT suppliers should conduct technical analysis to determine whether any given open source product complies with relevant open standards.

As a general rule, demonstrated compliance with open standards should be required when an OSS product needs to:

- Transfer data or documents to/from other applications;
- Store data or documents; and
- Interoperate with external applications via networks or other interfaces.

Industry standards and proprietary standards

Some standards do not strictly qualify as open standards but operate as de facto industry standards owing to widespread usage or broad industry support. The .DOC format of the Microsoft Word word-processing application is a well-known example. Some de facto standards are openly available to all vendors (industry standards) while others are only available under licence from the originating vendor (proprietary standards).

In general, industry standards and proprietary standards differ from open standards in a number of ways:

- Detailed technical specifications for document formats, data formats and communications protocols may not be clearly or openly defined;
- Other industry stakeholders (vendors, users) have minimal influence over the standards; and
- Originating vendors may introduce barriers (software patents or related intellectual property claims) to ensure that competitors are legally encumbered or locked out of the market.

Economic impact of open standards

The overlap between open source and open standards is generally stronger than the overlap between proprietary software and open standards.

Proprietary software vendors do not always adopt open standards for communications, data or document format protocols. By avoiding open standards, the vendor can make the process

of switching to alternatives more difficult. To move to an alternative provider, an agency must cover the cost of data and document migration, interoperability testing and re-training.

This is a standard lock-in scenario. In the formulation of vendor termination strategies, agencies should take this cost into account as part of the overall true cost of that ICT purchase.

Open standards can lead to the commoditisation of a particular technology. In such cases, vendors may experience reduced profit margins compared to transactions with clients that use proprietary products. Open standards tend to open competition and accelerate progress within a market segment due to additional competition.

An example of commoditisation can be seen in the personal computer (PC) hardware industry. Over the past decade, PC prices have fallen by 90% in real terms[13]. Over the same period, there was a 100-fold increase in PC power, features and capacity[14]. This price/performance boost came about due to intense price and feature competition among PC vendors and increased economies of scale brought about by standardisation of components. Such competition reflects greater use of open standards and industry standards in PC hardware, allowing all vendors to compete on equal terms.

Government agencies can use such models to identify ICT product segments where standards-based competition is evident. These segments may offer higher levels of product interoperability and greater choice in component procurement, leading to more competitive pricing. This in turn reduces system interoperability risks for the agency.

Open systems

In its formative years, the ICT industry was far less open. There were few mutually agreed standards. Many products lacked any facilities for data transfer or document interchange. Communication between products from disparate vendors was usually impossible because network protocols were all proprietary. As a consequence, users suffered from unnecessarily high prices.

The situation started to change in the 1980s. Computer users recognised that this lack of interoperability came at a cost for their organisations in terms of lost functionality, lost opportunities and maintaining disparate silos of technology. Users began to exert pressure on vendors to create open protocols for data, documents and networking. The aim was to improve interoperability between systems from different vendors.

At the same time, there was a push to encourage vendors to adopt standards-based programming languages and standards-based application programming interfaces (APIs) at the operating system level. The result was increased adoption of open standards. One example was the way operating system vendors embraced the Portable Operating System Interface (POSIX) standard, later known as IEEE 1003.1. POSIX was exemplified by the Unix operating system platform.

Many large organisations, government agencies and research establishments moved their core computing functions onto platforms that complied with these standards. This encouraged more vendors to embrace the same standards. These vendors produced POSIX-compatible systems interconnected on TCP/IP networks. Collectively such technologies were known as 'open systems'.

The move to open systems encouraged increased competition in the ICT industry. Functionality and the value of computer technology improved, although there were continued impediments to price competition and interoperability. The main problems arose when vendors began to deviate from agreed POSIX specifications. Many created

[13] Based on analysis comparing advertising in Australian Personal Computer magazine from December 1993 and December 2004.

[14] Based on analysis comparing advertising in Australian Personal Computer magazine from December 1993 and December 2004.

API extensions that were only available on their platform, resulting in increasing platform divergence and incompatible proprietary implementations of Unix. Vendors began using API differentiation as a marketing advantage to encourage users to select their platform over those of their competitors'. Independent software vendors (ISVs) also started to write software enhancements that were unique to specific platforms. This diminished interoperability between so-called open systems platforms.

Part of the original vision of open systems was the ability for a customer to switch from vendor to vendor easily and cost-effectively. However, while the open systems movement achieved significant advances, it never fully delivered the level of user control it originally promised.

One of the repercussions of this sequence of events was the growth of open source software. This came about because open systems failed to deliver a totally open, interoperable universal platform that offered a level playing field. Another factor was the history the Unix operating system. Originally developed by research and education institutions as a platform to train computer scientists, over time Unix moved from an open and freely accessible codebase to one closed off from computer science researchers and programmers.

The General Public Licence (GPL), the most prominent open source licence in use today, was devised as a way to prevent the future closure of product codebases. One impact of the GPL is that it removes the possibility of proprietary extensions to a codebase. Any extensions by one vendor must be legally made available for other vendors to adopt and resell.

While it is not impossible to convert a GPL-licensed product into a non-interoperable version, there is no financial gain for the vendor in doing so. The GPL licence removes the incentive to use product differentiation for financial gain by requiring that any differentiated code must be made available back to the open source community. As a result, vendors need to use quality of service and support to differentiate their value to customers.

Appendix C: Open source software packaging

If an agency is considering an in-house sourcing process, it needs to understand a number of complexities and issues involved in open source software packaging.

The process of installing and managing open source software varies greatly. Many of the popular packages are available on multiple computer platforms. The form of installation packaging usually depends on what installation formats are common on those platforms.

For example, versions of popular open source software designed to run on computers running Microsoft Windows are often packaged in a standard Windows installer utility. The software is usually compressed into a ZIP file or self-expanding executable file (EXE). Installation of this type of software is little different to installing traditional proprietary software. Most competent users would have few problems performing the installation.

However, a lot of open source software is available in platform-neutral archives, often in the form of compressed archive files. This is particularly common for applications that are developed as combinations of scripts and script libraries. Examples include many of the web-based applications such as groupware, content management systems and online database interface applications.

Server-based systems generally require the existence of various database, web and application servers prior to installation. They are more complicated to install and configure. Many require direct manipulation of system configuration files during the installation process. Agencies should make sure they have staff who are experienced in the technical configuration and management of software systems before undertaking such deployments.

Most mainstream open source applications are available in software bundles called packages. Packages involve a very different process of deploying and managing software.

For example, packages are very common on Linux and other open source platforms and most software available on the Linux operating systems is available as a package. A Linux distribution may contain between 500 to 5,000 such packages, each designed to deliver specific operating system functions. The whole Linux platform (kernel, libraries, components) and all applications are installed as packages with most packages including only files and modules directly related to that particular piece of software.

Complicating the picture is the fact that any particular package may require the existence of other software packages. This creates a hierarchy of interlinking dependencies where other packages may be required before a particular software package can be installed successfully. In the Linux world, engineers have produced a technical framework to safely

and easily manage this complexity, meaning users managing Linux systems need only install the single software package of interest. The technical framework tools determine what other system components are necessary then notify the user about the requirement and wait for acknowledgment before fetching these components from either a local or trusted remote software repositories. The Linux tools then install all the prerequisite packages, along with the main application the user had requested.

Agencies using in-house procurement procedures need to be aware of any additional software components that may need to be introduced onto the agency's computer systems to complete an OSS solution. Such software components need to be included in risk assessment.

For your notes:

A Guide to Open Source Software for Australian Government Agencies

Designed and Typeset by Levitate Graphic Design