Open Source Software for Government Assessment Model

February 2011

Aim

1. This document presents an assessment model for selecting Open Source software for use across Government, and the wider UK public sector.

2. It is presented in recognition that open source software is underused across Government and the wider public sector, despite the current and previous administrations intention to promote its use. It aims to address the lack of experience or understanding of open source software and its ecosystem of development and support.

3. This assessment model is developed in conjunction with IT systems integrators and suppliers to Government. This means this model is not in conflict with their internal assessment and selection processes.

4. The assessment of open source software suitability for Government use should be no different to the assessment of commercial closed software. However, there are specific areas where open source software is different, and this model reflects that.

5. This model cannot be overly prescriptive because the risk associated with software selection must remain with IT systems integrators and suppliers.

Context

1. The Coalition Government believes Open Source Software can deliver significant short and long term cost savings across Government IT.

2. Typical benefits of Open Source include lower procurement prices, no license costs, interoperability, easier integration and customisation, compliance with open technology and data standards giving autonomy over your own information and freedom from vendor lock in.

3. OSS is not currently widely used in Government IT, and the leading systems integrators for Government Departments do not routinely consider open source software for IT solution options, as required by existing HMG ICT policy.

© Crown Copyright

4. There are significant and wide ranging obstacles to Open Source in Government. Some of these are lack of procurement guidance, resistance from suppliers, concerns about license obligations and patent issues, and a lack of understanding of open source maturity and its development ecosystem.

Feedback

Please provide feedback to:

Tariq Rashid, Home Office, tariq.rashid@homeoffice.gsi.gov.uk

© Crown Copyright

1. Open Source Software for Government Assessment Model

Principle	Positive	Intermediate	Negative
Proven	• There are many real world examples of the software in use. • There are existing uses of the software in Government.	• There are some real world examples of the software in use. • The software, though in use, has not been used in Government or the wider public sector.	• There are no real world examples of the software in use. • There are only non-critical uses of the software.

	• The software is proven in heavy use with respect to:
	  ○ user demand
	  ○ performance
	  ○ data size and processing
	  ○ geographical reach.
		• The software is proven under moderate demand.	• The software has not been demonstrated in the real world to meet significant user demand, performance, data size and processing, or geographical reach.

	• The software has been successfully used over many years.	• The software has been successfully used over a few years.	• The software is new and has not been proven over time.
Support	• Support exists at several levels, including for mission critical services.	• Software is supported but not for mission critical services.	• No support arrangement.
	• Support has been proven over a number of years. • Support has been proven over a range of sectors, including demanding sectors such as	• Support is new and has not been proven over a number of years. • Support has only been proven over a	• No support.

© Crown Copyright

	finance and Government.	limited range of sectors.	
	• Software support includes bug fix and code change capability, as well as configuration support.	• Support capability only covers configuration support.	• No support.
	• Support supplier is the same, or closely aligned to, the software developers.	• Support supplier is not the same organisation as the software developer, nor does it have links to those developers.	• No support.
Security	• Changes to software are strictly assessed, audited and controlled through mature governance. This is the same as for well governed commercial software organisations.	• Changes to the software are limited to selected developers who have proven their ability to • Ad hoc contributions to the software are channelled through these selected developers for assessment and approval.	• Changes to the software are managed by s small inexperienced set of developers. • Changes to the software are subject to minimal, or no, assessment and governance.
	• Software is subject to regular security and penetration testing undertaken by the developer. Outcomes drive security fixes. • Software is penetration tested and accredited by an independent and trusted third party. Accreditation covers change control and governance, not just testing the resultant software.	• Software is regularly security tested by 3rd parties, with results published.	• No software security testing.
	• The software supplier is fully open and transparent about publicising known vulnerabilities and exploits. Impact assessment of vulnerabilities and exploits is issued. • All known vulnerabilities are acknowledged by	• The software supplier selectively publicises vulnerabilities and exploits. • The supplier does not acknowledge all known vulnerabilities.	• The supplier does not discuss vulnerabilities or exploits. • The supplier does not acknowledge vulnerabilities discovered by the wider community.

© Crown Copyright

	the supplier.		
	• The time to fix vulnerabilities is minimal. • All known vulnerabilities are fixed.	• The time to fix is not moderate. • Not all known vulnerabilities are fixed.	• Time to fix is long, or no fix is issued. • No management of vulnerabilities exploits or fixes.
	• Software is designed and developed to be secure.	• Normal software design and development practises.	• No software design and development discipline.
Software Development	• Software development is well managed and governed. • Software development is open to user contributions, but these are assessed and only enter the code in a well managed and governed manner.	• Software development is loosely managed, with change control but no coherent direction or quality criteria. • Software development is arbitrarily open to contributions from the user community but there is no formal assessment or governance around this.	• Software development is undisciplined, not following any process or governance. • Software development is not open to public contribution.
	• Software issues and bugs are managed by a bug tracking and resolution system that is transparent and has public visibility.	• Software issues and bugs are managed internally, with no public visibility.	• There is no coherent approach to tracking issues or bugs to resolution.
	• Software development is fully auditable for code changes, their source, and the reason for change.	• Software development is change controlled but with limited audit for code changes, their source, and the reason for change.	• Software development is not auditable.
	• Software is developed with full understanding and tracking of incorporated code from 3rd parties, patents and inherited license obligations.	• Software is developed in a fairly well controlled manner to provide confidence, but not guarantees, with respect to incorporated code from 3rd parties, patents and inherited license obligations.	• Software is developed with no understanding of incorporated code from 3rd parties, patents and inherited license obligations.

© Crown Copyright

IT Integrator or Supplier	• Integrator understands open source software and its ecosystem.	• Integrator understands open source software and its ecosystem only enough to make occasional use of software, primarily imitating other integrators.	• Integrator does not understand open source software and its ecosystem. and has no intention to use it strategically.
	• Integrator's legal and commercial units understand open source software and are fully engaged with it business as usual.	• Integrator's legal and commercial units partially understand open source software and only engage with it tactically and by exception.	• Integrator's legal and commercial units do not understand or engage with open source software.
	• Open source software plays a primary tier role in the integrator's strategic vision and approach to IT solutions and services. • Integrator maintains permanent internal expertise for strategic open source software.	• Open source software does not play a primary tier role in the integrator's strategic vision and approach to IT solutions and services. It only plays a secondary role where it is used tactically or by exception. • Integrator's internal expertise is incidental. Integrator buys in open source expertise on a temporary and case by case basis.	• Open source plays no part in the integrator's strategic vision or approach to IT solutions. • Integrator has no internal expertise in open source software, and does not engage temporary expertise.
	• Integrator has established channels to support and maintenance for open source software. • Integrator also has partnerships with design and integration specialists for specific open source software.	• Integrator engages channels to support and maintenance for open source software on a case by case basis, and often by exception. • Integrator only forms partnerships with design and integration specialists for specific open by exception.	• Integrator has no channels to open source support and maintenance. • Integrator has no partnerships with open source software specialists.
	• Integrator has proven successful experience of open source software, over a wide range of solutions and services, including mission critical.	• Integrator has some proven successful experience of open source software, over a limited set of solution and service types.	• Integrator has no proven experience of open source software as part of its solutions and services.

© Crown Copyright

	• Integrator is capable of managing open source software which has been modified to better meet the customer's requirements. The modification is undertaken internally or through partnerships with 3rd party specialists.	• Integrator does not manage modified open source software, only using well known releases from upstream support suppliers and developers.	• n/a
Legal & Indemnity	• Customer is shielded, or indemnified, from legal action related to patents, licenses or other issues by the software support vendor or the integrator. • The open source software has successfully defended against legal action, with respect to copyright, patents or other issues, in a comparable jurisdiction, thus setting a legal precedent.	• Customer has to undertake its own due diligence for legal liability around open source software, because it cannot be obtained from the software support vendor or the integrator. In the event of legal action, it must defend itself. • The open source software is subject to ongoing but as yet unresolved legal action, or is the subject of negative commentary from commercial rivals. Note that this is common in the software industry, with legal action and commentary used for competitive advantage and to influence customer behaviour.	• Customer cannot obtain any assurances around legal liability for the open source software in question. • The open source software is failed in defending against legal action, with respect to copyright, patents or other issues, in a comparable jurisdiction.
License	• Software license allows use for primary requirement, and also for moderately varying requirements. • Software license allows subsequent reuse across Government.	• Software license only allows use for specific requirement. Use for varying requirements is not permitted. • Software license does not allow reuse across Government.	• Software license forbids use for, or is incompatible with, primary requirement.
	• Software license places no onerous obligations on customers, subsequent to its use. GPL, MIT and BSD and similar licenses fall into this category, which is the majority	• Software license places some obligations or constraints on customers, subsequent to its use. Licenses which require the public release of sensitive code fall into this category, but this only applies when GPL style licensed code is integrated with sensitive code at the source code level.	• Software license places onerous obligations on customers, subsequent to its use. This is rarely true for common open source software.

© Crown Copyright

© Crown Copyright