Guidelines on Open Source Software in Public Procurement

01. Scope and Applicability

These guidelines apply to the public sector in the procurement process, including the stages of contracting and contract management. The guidelines address specifically the procurement of Open Source Software and compare the same with the procurement of Proprietary Software.

02. Purpose

The aim of this document is to facilitate the identification of tender requirements and the corresponding evaluation method of open source software. It also addresses legal issues related to Open Source Software with reference to contractual provisions and a governance framework for the management of the related licences at contract management stage.

03. Use of the Guidelines

This document is intended to guide Contracting Authorities in assessing Open Source Software and Proprietary Software in the procurement process. It is not to be interpreted as a mandatory or exhaustive guide to take into account in the procurement process. Contracting Authorities are expected to review the validity or otherwise of these guidelines in each procurement process. This document is not to be construed as legal advice.

04. Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Requirement</td>
<td>Means the requirement of the Contracting Authority to have rights to use the software being procured for a particular purpose including but not limited to redistribution and carrying out modifications to the software.</td>
</tr>
<tr>
<td>Contracting Authority</td>
<td>Shall have the same meaning as assigned in Public Procurement Legislation.</td>
</tr>
<tr>
<td>Functional Requirement</td>
<td>Means the requirement of the Contracting Authority defining a particular function of software or part of the software to accomplish an expected result. It covers input, behaviour, processing and output but excludes features or characteristics of the software including but not limited to security, scalability, reliability which do not denote a particular function of the software.</td>
</tr>
<tr>
<td>Generic Software</td>
<td>Also referred to as “off-the-shelf”, means ready-made software.</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technologies.</td>
</tr>
<tr>
<td>Intellectual Property Rights or IPR</td>
<td>Includes, without limitation, patents, trade marks, registered designs, copyright, design rights and any other intellectual or industrial property rights.</td>
</tr>
<tr>
<td>Invitation to Tender</td>
<td>Means the document issued or to be issued by a Contracting Authority inviting the public to submit proposals for the provision of works, supplies and/or services.</td>
</tr>
<tr>
<td>Open Source Software or OSS</td>
<td>Means software for which the underlying programming code is available to users to read, make changes to it, and build new versions of the software incorporating their changes. In addition to access to the source code, the Open Source Initiative (OSI) refers to ten (10) criteria that define Open Source Software available at the OSI web site: <a href="http://www.opensource.org/docs/osd">http://www.opensource.org/docs/osd</a></td>
</tr>
<tr>
<td>Open Source Software Licence</td>
<td>Includes any such licence listed at</td>
</tr>
</tbody>
</table>
Guidelines on OSS in Public Procurement

Open Specifications (informally referred to as Open Standards)

Are formalised specifications which within the context of public services delivery, are characterised by the following features:

- All stakeholders have the same possibility of contributing to the development of the specification and public review is part of the decision-making process;
- The specification is available for everybody to study;
- Intellectual property rights related to the specification are licensed on fair, reasonable and non-discriminatory (FRAND) terms or on a royalty-free basis in a way that allows implementation in both Proprietary and Open Source Software.

Proprietary Software

Also referred to as “closed source”, means software which is not OSS, copyrighted and traditionally governed by extensive terms and conditions.

Proprietary Software Licence

Includes licences referring to:

(i) Prohibitions for modification to the original version of the software;
(ii) Prohibitions on redistribution of the original and/or any modified versions of the software;
(iii) A fee for use;
(iv) Restrictions on use e.g. restricted to “non-commercial use”;
(v) Prohibitions or limitations on assignment and transfer.

Public Procurement Legislation

Means the Public Procurement Regulations (promulgated by virtue of Legal Notice 296 of 2010).

Technical Requirement

Means the requirement of the Contracting Authority for the software to be capable of particular attributes including but not limited to modifications in order to work with other software.

Tenderer

The person submitting a response to the Invitation to Tender.

05. Overview

This document identifies various stages in the procurement process and addresses the issues that need to be considered by Contracting Authorities in relation to Open Source Software. The stages referred to are

1. Preparation of a Business Case and Research,
2. Deciding on the Sourcing Strategy,
3. Drafting Tender Requirements,
4. Evaluating Open Source Software and Proprietary Software, and
5. Contractual Considerations, including contract/licence management.

06. Preparation of Business Case and Research

There are different drivers which create a business need for change. The drivers and the corresponding business needs need to be documented within a comprehensive business case. In addition, a proper research exercise should enable the identification of opportunities for

- Capitalising on existing investment,
- Future re-use opportunities,
- Need for source code access,
- Interoperability opportunities,
- Adopted and new standards,
- Innovative technologies,
- Market conditions and
- A cost estimate of direct and indirect costs as may be required to calculate the total cost of ownership (TCO).
Appropriate research and/or consultancy also facilitate the procurement process, aiding Contracting Authorities to define accurate business requirements. The outcome of the research should be documented and attached to the business case as it may be required to justify related business requirements.

07. Deciding on the Sourcing Strategy

Open Source Software can be acquired by Contracting Authorities internally, free of charge and without entering into a contract, by downloading the software from online forges and repositories. However, the Contracting Authority needs to determine the suitability of the software against its requirements. In addition, if the Contracting Authority does not intend to use the software as is, it needs to have the internal capability and resources to perform related services [such the carrying out of modifications] itself. If such capability and resources are absent it may opt procuring such related services separately.

Where Contracting Authorities opt to procure software, requirements need to be identified for Tenderers to propose software (whether Open Source or Proprietary Software) to meet the requirements. Contracting Authorities may include software and related services in the same procurement process.

08. Drafting Tender Requirements

Once a decision is taken by the Contracting Authority to procure software from the open market, the Contracting Authority needs to consider Open Source Software alongside Proprietary Software. As stated in the Government of Malta Open Source Policy and Directive (GMICT P 0097 and GMICT D 0097), Contracting Authorities must ensure a level playing field between Open Source Software and Proprietary Software.

Contracting Authorities cannot refer directly to Open Source Software as a requirement in the Invitation to Tender. Open Source Software (as with Proprietary Software) can only be considered in the procurement process if it meets the business needs. Business needs are transposed in the Invitation to Tender through Business, Technical and Functional Requirements which may indirectly call upon the properties inherent in Open Source Software.

What are the properties of Open Source Software and how can a Requirement indirectly call upon such properties?

When a Contracting Authority assesses the technical and functional specifications of Open Source Software and Proprietary Software, both types of software are evaluated on a par level. There are no Technical and/or Functional Requirements that can specifically distinguish between Open Source and Proprietary Software. Business Requirements are what can distinguish between the two. In fact, when a Contracting Authority assesses the license terms and conditions of Open Source Software and Proprietary Software, a Contracting Authority may note that the Open Source Software license gives more flexibility in the use of the software.

The flexibility of Open Source Software licenses is inherent in the properties listed by Open Source Initiative (OSI) in the Open Source Definition (OSD):
The Open Source Definition

Introduction
Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups
The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral
No provision of the license may be predicated on any individual technology or style of interface.
All ten criteria apply to Open Source Software. Terms of licences that do not provide all such ten criteria would refer to Proprietary Software. Proprietary Software may, however, provide for some (but not all) of the criteria. Therefore, for instance, in the case of Generic Software having potential for re-use, a Contracting Authority may request the right to re-use the software being procured in subsequent projects within the Government of Malta without limitation as a Business Requirement. On the one hand, the licensor of Proprietary Software may commit itself to allow such right of re-use without limitation. On the other hand, Open Source Software licences are a priori (by default) compliant with a request for re-use without limitation.

The re-use requirement is one example whereby a Business Requirement calls upon the properties of Open Source Software. Other examples are listed in the ‘Guideline on Public Procurement of Open Source Software’ issued by IDABC in May 2010 as follows:

- The software may be used for any purpose as the public agency does not want to be restricted in how it can use (or allow others to use) the software.
- The public agency or a third party of its choice may study the source code as the public agency wants to be sure of the functioning of the software; alternatively, the public agency may require that any member of the public can study the source code, in order to promote transparency of government processes, or enable other parties to provide support and training associated with the software.
- The public agency or a third party of its choice may modify the software as the public agency does not wish to be dependent on the original vendor for bug-fixes, adaptations and other modifications.
- The public agency can distribute the software, with source code and modifications, to anyone of its choice and provide recipients with the same abilities to use, study, modify and redistribute because the public agency needs to ensure that citizens and firms and other agencies that access its services using the software or variants of the software do not need to become customers of the original vendor in order to do so; for example, a national administration may wish to be able to pass on the software, without extra costs, to other administrations at the local, regional, national or European levels.

The Business Requirements define the rights of use that the Contracting Authority wishes to procure. Although all of the Business Requirements listed above (collectively), indirectly, refer to Open Source Software, certain rights of use inherent in the same Business Requirements can be provided by Proprietary Software. Hence, the requirement in Public Procurement Legislation not to refer directly to a specific category of software (such as Open Source) – because if a Contracting Authority requires only a number (but not all) of the Business Requirements, Proprietary Software may compete with Open Source Software.

### 08.1 Comparing Open Source Software and Proprietary Software:

The below table summarises the salient features of Open Source Software and Proprietary Software with a view to point out differences and similarities. The table is based on A Guide to Open Source Software for Australian Government Agencies published by the Australian Government Information Management Unit.

<table>
<thead>
<tr>
<th>Source Code</th>
<th>Open Source Software</th>
<th>Proprietary Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Source code is made available for anyone free of charge to view, vet and modify.</td>
<td>Costs and other restrictions may be imposed to provide access to view the source code.</td>
</tr>
<tr>
<td>Capital Expenditure</td>
<td>Source code and object code available free off charge. Proponent may charge solely for the medium in which the source code and/or object code.</td>
<td>Source code and object code are usually chargeable in addition to the related services.</td>
</tr>
<tr>
<td></td>
<td>Related services such as customisation (software development).</td>
<td></td>
</tr>
<tr>
<td><strong>configuration, integration, implementation, training, translation may be</strong></td>
<td><strong>chargeable.</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Maintenance and support are also chargeable (whether a supplier is</strong></td>
<td><strong>engaged or internally if resources are allocated)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TCO costs need to be considered – including but not limited to the</strong></td>
<td><strong>underlying infrastructure.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Customisation</strong></td>
<td><strong>Customisation may be carried out in-house or outsourced to a supplier.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>In case of restrictive licences,¹ customisations must be made available if the software is re-distributed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customisation is usually carried out by the supplier of the software.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPR for customisations is usually kept by supplier of the software.</td>
<td></td>
</tr>
<tr>
<td><strong>Support</strong></td>
<td><strong>Usually developed in communities that may be able to provide professional support in case of problems with the software. Professional support varies from one community to another. As an indicator: “An open source software community with an active and diverse membership, a broad user base, a good governance structure and regular updates is likely to be more responsive to user requests.”²</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Supplier of the software should provide information on the manufacturer of the software, software life-cycles, governance procedures and product roadmaps.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support may be provided by the supplier of the software.</td>
<td></td>
</tr>
<tr>
<td><strong>Innovation</strong></td>
<td>Various sources available within the community to contribute to the creation of the software which encourages innovation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Innovation is driven by the manufacturer that creates the software.</td>
<td></td>
</tr>
<tr>
<td><strong>Intellectual Property</strong></td>
<td>IPR is usually non-negotiable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPR can be negotiated.</td>
<td></td>
</tr>
<tr>
<td><strong>Re-Use Opportunity</strong></td>
<td>Promotes Re-Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re-distribution of modifications may be imposed in the case of restrictive licences. In such cases the governance aspect is critical.³</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Re-use is generally limited. Any requirement for re-use needs to be specifically negotiated.</td>
<td></td>
</tr>
<tr>
<td><strong>Standards</strong></td>
<td>Generally adheres to Open Specifications. In cases where Open Specifications are not adhered to, Open Source Software may be customised to adhere to Open Standards.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nowadays, manufacturers are generally aware of Open Specifications. In cases where Open Specifications are not adhered to, the manufacturer must be requested to customise the software to adhere</td>
<td></td>
</tr>
</tbody>
</table>

¹ For an explanation of restrictive Open Source Licences see Section 10.1.1 on Intellectual Property Rights.
³ For an explanation on governance in relation to Open Source Software Licences see Section 10.2 on Contract Management.
<table>
<thead>
<tr>
<th>Specifications</th>
<th>to Open Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lock-in</strong></td>
<td>Both Open Source and Proprietary Software may drive Contracting Authorities to be locked-in. Lock-in is dependant mainly, but not limitedly, to lack of adherence to Open Specifications and lack of support options.</td>
</tr>
<tr>
<td><strong>Release Management</strong></td>
<td>Software patches, software updates and product releases should be aligned with Government governance process, including evaluation processes, testing procedures, patch management and technology life cycle.</td>
</tr>
<tr>
<td><strong>Restrictions on Use</strong></td>
<td>If the software is used as is, generally, no restrictions apply. In case of distribution, obligations are triggered on the distributor including but not limited to an obligation provide access to the source code under the terms and conditions of the licence. In the case of modifications, Open Source Software Licences that are restrictive require the user to make available modifications to the source code available in the case of distribution. Conditions within licenses may restrict the use of the software including but not limited to conditions referring to hardware, site, user, frequency, dates and others. These terms are, however, negotiable.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>All software may be vulnerable to attacks. Therefore security risks in relation to the software need to be identified (by the Contracting Authority or through advisories) and assessed (by the Contracting Authority). In addition, appropriate patching and security procedures need to be adopted to mitigate security risks.</td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>Requires Administrative and End User Training even if the upgrading from one version of a solution to another.</td>
</tr>
<tr>
<td><strong>Liabilities and Warranties</strong></td>
<td>Generally, Open Source Software Licences exclude liabilities and warranties to the maximum extent permitted by law. A Contracting Authority would need to procure such liabilities and warranties from a supplier at a cost. Liabilities and warranties are negotiable.</td>
</tr>
</tbody>
</table>

09. **Evaluating Open Source Software and Proprietary Software**

09.1 **Business, Technical and Functional Requirements**

Both Open Source Software and Proprietary Software can be checked against the Business, Technical and Functional Requirements. Where the Contracting Authority justifies the need of such requirements, the evaluation entails merely an assessment on whether the software proposed by the Tenderer meets the requirements set by the Contracting Authority in the Invitation to Tender i.e. whether it meets minimum requirements. In such cases any difference between Open Source Software and Proprietary Software may feature, financially, through the Total Cost of Ownership. This approach, however, assumes that a Contracting Authority is awarding a public contract on the basis of the lowest price offered compliant with the tender specifications. In cases where the Contracting
Authority sets tender requirements that are desirable, the software proposed by the Tenderer may be evaluated and the public contract awarded on the basis of the most economically advantageous offer.

09.2 Total Cost of Ownership:

The total cost of ownership (TCO) is a term used to determine the economic value of an investment. It is extensively used in procurement to denote value for money. TCO includes direct and indirect costs.

Direct costs are identifiable at the moment of acquisition. They are fixed at the moment of acquisition but variable in the long run. Indirect costs are not easily identified given that they refer to costs incurred by the Contracting Authority as a result of the acquisition. Such costs are generally not fixed and do not relate specifically to the acquisition (but various acquisition). Indirect costs may be grouped into a fixed cost, on the basis of estimates, to facilitate evaluation.

A list of examples of what constitutes direct and indirect costs is referred to below. The list in not exhaustive and there may be variances.

Direct costs
- Software and Hardware Installation;
- Systems Integration and Data Migration;
- Software Customisation;
- Software and Hardware Maintenance;
- Training;
- Analysis and Research; and
- Warranties and Licenses.

Indirect Costs
- Infrastructure Costs, including upgrade related to bandwidth requirements;
- Insurance Costs (Risk insurance against trademarks and patents infringements);  
- Office Space;
- Power and cooling;
- In-house support (including developers, e.g. to facilitate integration, to accelerate timeframes, to contribute back to community);
- Failure Expenses;
- Performance Upgrades;
- Auditing;
- Exit Costs; and
- Re-tendering/Transition costs/Decommissioning

Such costs may be incurred at acquisition stage or, subsequently, during the term of the contract (i.e. length of investment). Although incurred in different stages, the Contracting Authority needs to take into account all costs possible at acquisition stage, possibly by limiting variables during the term of the contract e.g. agreeing to fixed costs over the term.

10. Contractual Considerations:

10.1 Contractual Provisions:

The more relevant contractual considerations in relation to Open Source Software concern:

10.1.1 Intellectual Property Rights

Open Source Software Licences grant the basic freedoms to users to copy, modify and distribute the source code and object code. In the case of modifications, a number of Open Source Licences require that the modifications to the software are made available to the development community under the same terms and conditions of the original Open Source Licence. These licences are referred to as “restrictive” licences because they impose an obligation to allow copies, modifications and distribution

---

4 Refer to: http://www.osriskmanagement.com/insurance.html
to the “derived work” (i.e. the work produced from modifications to the Open Source Software). Other Open Source Licences are “permissive”, allowing the user to commercially exploit the modifications under a Proprietary Software Licence.

Hence, whereas restrictive Open Source Licences cannot be combined with Proprietary Software Licences, permissive Open Source Licences can be combined with Proprietary Software Licences. This implies that when a supplier proposes Open Source Software regulated by a restrictive Open Source Licence, the supplier must ensure that modifications or derivative works are capable of being distributed with and under the same terms and conditions applicable to the Open Source Software component. Likewise the contract must treat Open Source Software governed by a restrictive Open Source Software Licence separately by expressly stating the terms and conditions of the applicable Open Source Licence (which is restrictive) in a separate schedule of the contract. In case of permissive Open Source Software Licenses the modifications or derived works can be regulated by the standard terms and conditions listed by the Contracting Authority in the contract to regulate the ownership (including assignment) of Intellectual Property Rights created during the term of the contract.

Open Source Software Licences generally exclude warranties and liabilities. However, there is nothing prohibiting a supplier from agreeing to provide warranties ensuring that:

1. The Contracting Authorities has the all the rights necessary to use the software as intended and on a perpetual basis.
2. The Contracting Authority is indemnified against breaches of third party Intellectual Property Rights in the course of using the software provided, covering any losses, damages, costs or expenses or other liabilities incurred.

When procuring Open Source Software, Contracting Authorities need to set a time-frame within which the supplier is to provide a copy of or access to the source code on delivery of the Open Source Software and any modifications or derived works, at no additional cost. Escrow arrangements are not required in the case of Open Source Software given that the source code and object code are freely available.

10.1.2 Due Diligence

A supplier of Open Source Software would be required to warrant that the proposed software is compatible with any existing technologies of the Contracting Authority. Due diligence obligations on the supplier, ensure that the supplier proposes Open Source Software that has been evaluated and tested against the ICT infrastructure of the Contracting Authority. Effectively, due diligence carries forward the promises made at procurement stage.

10.1.3 Service Improvement

An obligation on the supplier to monitor and contribute to the developments of the Open Source Software proposed ensures that the Contracting Authority benefits off the more recent versions or updates to the software. The contract should cater for a process agreed to by the parties on how upgrades or updates shall be incorporated to proposed Open Source Software.

10.1.4 Maintenance and Support

In absence of any express provision, Open Source Software maintenance and support is dependent upon patches, updates and software product releases by the relevant Open Source Software community. Therefore, in procuring software and related services, the Contracting Authority has an interest to oblige the supplier to warrant that it will maintain and support the software throughout the contract term, irrespective of the existence of the community.
10.2 Contract Management

This section proposes a governance framework specifically for dealing with Open Source Software licences. It identifies the risks associated with the use Open Source Software Licences (not Open Source Software) and proposes mitigation to the risks identified. This table is based on *A Guide to Open Source Software for Australian Government Agencies* published by AGIMO in June 2011.

<table>
<thead>
<tr>
<th>Purpose for which OSS is required</th>
<th>Associated Risk</th>
<th>Risk Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use as is (without distribution or modification)</td>
<td>- Negligible</td>
<td>- Read, understand and ensure compliance with all licence terms and conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use as is (with distribution but without modification)</td>
<td>- Lack of distribution rights; - Obligations triggered upon distribution; - Liability and warranty risks assumed as distributor of the software</td>
<td>- Review licence to confirm distribution rights; - Read, understand and ensure compliance with all licence terms and conditions, including obligations triggered upon distribution. - Disclaim or other limit liabilities and warranties as distributor of the software.</td>
</tr>
<tr>
<td>Modify</td>
<td>- Lack of modification rights; - Conflict between arrangements made to carry out modifications to the software, implement and maintain and support and the licence terms; - Lack of compatibility between software licences of different components e.g. between restrictive OSS licences and Proprietary Software licences. - Lack of clear understanding on the use of the software – in case of distribution may trigger restrictive licence terms.</td>
<td>- Review licence to confirm modification rights; - Ensure that IPR of works is assigned to the Contracting Authority through public contract and contracts of employment. - Determine the applicable licences prior to distributing the packaged software. - Ensure that the use of the software is determined prior to distribution in order to identify the relevant obligations of the licence terms and conditions and comply with the same.</td>
</tr>
</tbody>
</table>

11. Related Documents

This document must be read in conjunction with the following GMICT publications:

- Open Source Software Policy (GMICT P 0097)
- Open Source Software Directive (GMICT D 0097)
- Open Standards Policy (GMICT P 0099)
- Open Standards Directive (GMICT P 0099)
12. Resources