Evaluation Criteria for Free/Open Source Software Products Based on Project Analysis
What is the article about

In this article, they give a systematic approach for supporting a decision to incorporate a particular FLOSS product into an enterprise.
In this article, they give a systematic approach for supporting a decision to incorporate a particular F/OSS product into an enterprise.
1. PROJECT CIRCUMSTANCES SUPPORTING A PRODUCT DECISION

What to look at before making a decision
- Similar selection process as COST software
- Functionality
- Source code inspection
- Determining the requirements on an F/OSS and evaluating the extent to which these requirements are actually met.
2. OUTLINE OF THE PRESENTED APPROACH
Figure 1. Overview of decision support approach
3. USAGE SCENARIOS FOR F/OSS

It shows possible usage scenarios for an F/OSS package.

Classical application scenarios for F/OSS are web, mail or file server, firewalls, or embedded systems.
It show possible usage scenarios for an F/OSS package.

Classical application scenarios for F/OSS are web, mail or file server, firewalls, or embedded systems.
4. REQUIREMENTS ON FREE/OPEN SOURCE SOFTWARE
Functional Requirements

1. Required Functionality Covered
   - The software comprises all the features it is supposed to.

2. Clear Direction of Product Evolution
   Recognisable
   - A clear "roadmap"
Technical Requirements

1. Target Platforms Supported
   - The software supports all the platforms in which it is intended to be used

2. Reliability
   - Robust with respect to expected and unexpected failures.

3. Maintainability
   - Easy to modify, improve performance
Organisational Requirements

1. Community Exists
   - Persons who can be contacted to exchange information about the software.

2. Product Evolution
   - The software is being adapted to the changing environment.

3. Sufficient Support Available
   - Contacting someone that can help if technical difficulties arise in working with the software.
Legal Requirements

1. No Copyleft Effect for Add-ons or Combinations
   - The licence does not oblige the company to licence own add-ons under the same licence as the F/OSS.

2. No Liability for Third Party Code
   - The company is not liable for the foreign code parts.

3. No Patent Infringements
   - Demands that the use of the F/OSS in question does not collide with any patent restrictions.
Economical Requirements

1. Sustainability of the Usage of the F/OSS
   - Long-term availability of product, support and maintenance.

2. Protection of Investment for Migrating to the F/OSS Product
   - The financial cost and time effort for migrating to the F/OSS product has to be prepaid.

3. Quick Availability of the F/OSS
   - The F/OSS is ready for use easily and in a short time.
Political Requirements

1. Possibility for Influencing Further Development of the F/OSS with Respect to Individual Needs
   - Customized according to company-specific requirements; the community is willing.

2. Publicity, Marketing Effects Through Usage of and Participating in F/OSS
   - It is publicly known that a company or organization participates in an open source project.
<table>
<thead>
<tr>
<th>Usage scenario</th>
<th>Related requirements of Section 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use F/OSS as a platform for a mission critical process. (The F/OSS is</td>
<td><strong>Functional</strong>: Functionality covered, clear evolution direction</td>
</tr>
<tr>
<td>existentially important for a company’s core process; every problem causes</td>
<td><strong>Technical</strong>: Target platform supported, reliability, maintainability</td>
</tr>
<tr>
<td>considerable costs.)</td>
<td><strong>Organisational</strong>: Maintenance active, sufficient support, long life existence</td>
</tr>
<tr>
<td></td>
<td><strong>Economical</strong>: Sustainability, flexible maintainability according to individual needs</td>
</tr>
<tr>
<td></td>
<td><strong>Political</strong>: Possibility for influencing further development according to individual needs,</td>
</tr>
<tr>
<td></td>
<td>decrease of proprietary dependencies, transparency over security</td>
</tr>
<tr>
<td>2. Use F/OSS with a long-term consideration. (The F/OSS is a part of the</td>
<td><strong>Functional</strong>: Clear direction of product evolution</td>
</tr>
<tr>
<td>technological strategy of an enterprise.)</td>
<td><strong>Technical</strong>: Target platforms supported, maintainability</td>
</tr>
<tr>
<td></td>
<td><strong>Organisational</strong>: Community exists, long life existence</td>
</tr>
<tr>
<td></td>
<td><strong>Economical</strong>: Sustainability, protection of investment</td>
</tr>
<tr>
<td></td>
<td><strong>Political</strong>: Possibility for influencing further development with respect to individual needs</td>
</tr>
<tr>
<td>3. Use F/OSS as a cost reduction model. (The F/OSS replaces a proprietary</td>
<td><strong>Organisational</strong>: Community exists</td>
</tr>
<tr>
<td>product and the license fee is saved.)</td>
<td><strong>Economical</strong>: Increasing know-how</td>
</tr>
<tr>
<td>4. Use F/OSS as exploration object, for example, for technology. (The F/OSS</td>
<td><strong>Functional</strong>: Required functionality covered, clear direction of product evolution</td>
</tr>
<tr>
<td>is used as a study object to learn about new technologies or paradigms.)</td>
<td><strong>Economical</strong>: Sustainability, protection of investment, cost reduction, division of development</td>
</tr>
<tr>
<td></td>
<td><strong>Political</strong>: Possibility for influencing further development with respect to individual needs</td>
</tr>
<tr>
<td>5. Use F/OSS as an exhibition prototype. (The F/OSS supports an exhibition of</td>
<td><strong>Organisational</strong>: Community exists</td>
</tr>
<tr>
<td>a product owned by the company or is exhibited as its own further development.)</td>
<td><strong>Economical</strong>: Flexible maintainability according to individual needs, quick availability</td>
</tr>
<tr>
<td></td>
<td><strong>Political</strong>: Publicity, marketing effects</td>
</tr>
<tr>
<td>6. Use F/OSS as a base line for further development and business model. (The</td>
<td><strong>Technical</strong>: Target platforms supported, maintainability</td>
</tr>
<tr>
<td>F/OSS is the base line for further proprietary development to be sold or to</td>
<td><strong>Legal</strong>: No copyleft, liability, patent infringements, reselling obligations</td>
</tr>
<tr>
<td>be subject of support contracts.)</td>
<td><strong>Economical</strong>: Flexible maintenance according to individual needs, increasing know-how</td>
</tr>
<tr>
<td></td>
<td><strong>Political</strong>: Transparency over security</td>
</tr>
</tbody>
</table>
7. Use F/OSS to bridge a temporary bottleneck. (The F/OSS fills a temporary gap in the migration from one software package to another.)

8. Use F/OSS for becoming independent of proprietary solutions and providers. (The F/OSS helps to reduce dependence on software product companies with big market power and low customer orientation.)

Functional: Required functionality covered
Technical: Reliability
Organisational: Sufficient support available
Economical: Flexible maintainability according to individual needs, quick availability
Political: Decrease of proprietary dependencies

Functional: Required functionality covered, clear direction of product evolution recognisable
Technical: Reliability
Organisational: Community exists, maintenance active, sufficient support available
Economical: Sustainability, flexible maintenance according to individual needs
Political: Possibility for influencing further development with respect to individual needs, decrease of proprietary dependencies
9. Use F/OSS to gain transparency concerning safety and security. 
(The F/OSS is used to become independent from software companies not trustworthy in security concerns.)

10. Use F/OSS for research purposes. 
(The F/OSS is used to support research activities.)

11. Use F/OSS as a CASE tool. 
(The F/OSS is used for developing software.)

- **Functional**: Required functionality covered
- **Economical**: Sustainability
- **Political**: Decrease of proprietary dependencies, transparency over security

**Technical**: Reliability

**Organisational**: Sufficient support available

**Economical**: Flexible maintenance according to individual needs

**Political**: Possibility for influencing further development with respect to individual needs

**Functional**: Required functionality covered, clear direction of product evolution recognisable

**Technical**: Target platforms supported, reliability

**Organisational**: Community exists, maintenance active, sufficient support available

**Economical**: Sustainability, protection of investment, increase productivity, flexible maintainability according to individual needs, cost reduction, division of development costs

**Political**: Possibility for influencing further development with respect to individual needs, decrease of proprietary dependencies