Evaluation Criteria for F/OSS Products Based on Project Analysis

PART 2
Some criterions are abstract and hard to determine, others are easy to evaluate quantitatively.

This section focuses on issues about software projects relevant to evaluation of F/OSS products and that are investigable.
Functional Criteria

- Evaluation of the features of the software
- Can be determined easily through feature lists if they exist
- If none exist, software must be tested extensively for evaluation
- To determine the functionality of a F/OSS product exhaustively requires different methods for different types of projects and there is no general solution on how to do this
Technical Criteria

- Actual number of bugs, depends on the developers keeping a public bug list. The bug list is rarely a complete list of bugs.

- Number of open feature requests, either new or modified, that have not been implemented yet. Tracking software for requests usually gives status of open, assigned, realised or discarded.

- Code metrics; number of files, classes, avg length of functions, number of comments per code lines. Can be found with specialized tools.
Technical Criteria

- Frequency of changes; Information about how often project files are changed and how often the code is checked in to repository. The code repository is usually publicly available allowing scripts to do this evaluation.

- Dependencies on other software; The number of other software packages the software relies on, usually available in the readme or can be discovered quickly by compiling and running.
Organisational Criteria

- Number of developers; Usually available in the project page
- Number of testers; There is often a system for users to submit bugs, such as BugZilla. Find the number by counting the different identities that are not developers
- Number of users; Has to be estimated, by number of downloads, forum activity, etc
Organisational Criteria

- Development process patterns; The rules by which the project follows, usually available in the homepage or by request to project leader.

- Skills of the developer community; can be measured by letting an expert follow internal development discussions. If sufficiently skilled, the expert can usually determine the skill level of the developers involved by analyzing their discussions.
Legal Criteria

- Legal issues are connected to the license used, and can be determined rapidly if the software uses a common license.
- Copyleft effect of license; Whether combination software that includes the project must use the same license as the project.
- Liability when reselling; If a company selling a product that uses the project software is liable for errors in that software.
Economical Criteria

- Is there sufficient availability of development resources; Amount of infrastructure and resources available through companies participating in the project. Can generally be found through project home page

- Estimated Migration Efforts; A cost benefit analysis for how much it would cost for a company to switch to using the project software
Economical Criteria

- Estimated monthly costs; Considering maintenance or in-house support for running the software. Has to make analysis considering running effort and the cost for in-house maintenance and support for the project software.
Political Criteria

- Which well-reputed companies are involved; More companies supporting means a lower risk on relying on the software. Can usually be found in project home page and relevant IT press

- Which dependencies on other F/OSS exist; the more dependencies, the more complicated risk analysis becomes. Usually available in project homepage
In how many languages does documentation exist; The more languages supported and the better the documentation, the larger is the community and broader is the usage. Should be available on project homepage.

How is the climate in discussion forums; Constructive progress-oriented climate or flaming? The more progressive the attitude, the more lively the project can be deduced to be.
Political Criteria

- Are any large contributors about to be merged/bought in/insolvent; If large contributors are about to retire from the project it is considered less lively. One can find this by researching the supporting contributors.

- Are any large contributors about to have strategic or executive board change; New leadership might consider the project inappropriate for their directions. Found by researching supporting contributors.
Fulfilment of Requirements

- We now look at how to fulfill the requirements discussed
Fulfilment of Functional Requirements

- 1 – Compare list of required features and list of reportedly available features
- 2 – Check if features are actually implemented
- 3 – Check if non-listed features are actually not implemented
- 4 – Estimate cost of implementing requirements that are absent oneself
Fulfilment of Technical Requirements

- Number of bugs; Difficult to interpret, many bugs reported can mean more active/dedicated community rather than buggy software. Bug quality should be analysed before decision.

- Number of feature requests; More feature requests means more vivid community and usually that the software is satisfying (actually being used).
Fulfilment of Technical Requirements

- Code metrics; If many code lines per comment maintainability might be low. If functions have more than 100 lines on average code quality might be low and stability can be affected.

- Dependencies on other software; Lots of dependences means complex configuration. Not recommended.
Fulfilment of Organisational Requirements

- Community exists; Number of developers/users/testers has to be greater than zero
- Product evolution; Is the code actually changing over time? If not, product is not evolving
- Sufficient support available; If community exists, does documentation exist? What's the climate in discussion forums and mailing lists?
Fulfilment of Organisational Requirements

- Long Life Existence; Large communities and/or involvement of well-reputed companies means a great interest in the software, which signals long life

- Comptitable development process; Is the development process compatible with the organisation?
Fulfilment of Legal Requirements

- No copyleft effect for add-ons or combinations; Should check if associated software also has copyleft and how to avoid «license contamination»
- No liability for third party code; Determined by criterion «liability when reselling»
- No patent infringements; Difficult to check, but if large companies use they usually have done patent research and the software should be «clean»
Fulfilment of Economical Requirements

- Sustainability; If community is large and active then there is small risk software will die
- Protection of investment; If stable development circumstances exist, the investment is usually safe
- Increase productivity; Features should be implemented in the evolution of the software to increase productivity of software
Fulfilment of Economical Requirements

- Flexible maintainability according to individual needs; Is the skill and resources for further development adequate?
- Quick availability; Product dependencies and disturbing side effects give estimate of how fast the product is available
- Increasing know-how; If developers are knowledgeable the in-house staff can learn from the community
Fulfilment of Economical Requirements

- Cost reduction; Minimum of dependencies and a helpful community reduce costs
- Division of development costs; Not having to develop in-house but rather through community, so a vivid community is a positive factor
Fulfilment of Political Requirements

- Possibility for influencing further development with respect to individual needs; Community should be open for change and requests from product users. Check community for climate

- Decrease of proprietary dependencies; If community is skilled and has enough resources it should stay independent with the aid of well-reputed companies
Fulfilment of Political Requirements

- Transparency over security; Will give insight to source code and a readable design of the code
- Increase publicity, marketing effects; Involvement will increase if the product is well-known and reputable