

Use Cases: Open Source

Last update: 10/19/11

Use cases

- 1. We need to choose software solutions, potentially open source, for a project.**
 - a. Decision-making process
 - i. What is our current decision-making process?
 - ii. Do we want to formalize the process? Factors could include:
 1. Local context
 - a. Available development resources
 - b. Project timeframe
 - c. Nature of project (pilot, fundamental production service)
 - d. What's already being used, sunk costs
 - e. Local expertise
 2. Larger context - What are others using?
 3. Comparison to other options
 - a. Ability to meet project/system requirements
 - b. Cost (purchasing, initial customization, ongoing fees and costs)
 - c. Long-term considerations (maintenance, etc.)
 4. License terms
 - iii. Do we want to have a policy to preference open source software?
 - iv. What are the costs of using open source software?
 - v. What are the benefits of using open source software?
 - b. Documentation/communication of process and decision to provide rationale to others
- 2. We learn of an opportunity to participate in an existing project to collaboratively develop an open source software product.**
 - a. What are impediments to doing more of this?
 - b. What are the costs to us?
 - c. What are the benefits to us?
 - d. What are the characteristics of software, its user community or collaborators that lead to successful open source software collaborations?
- 3. We see an opportunity to initiate the development of a collaborative open source software product.**
 - a. What are impediments to doing more of this?
 - b. What are the costs to us?
 - c. What are the benefits to us?
 - d. What are the characteristics of software, its user community or collaborators that lead to successful open source software collaborations?
- 4. We have locally-developed software that could be made open source**
 - a. Process
 - i. Decision-making process (whether or not to open source a product)

1. What are the characteristics of software or its user community that makes it amenable to being open sourced?
 - ii. Approval process (if applicable)
 - iii. License selection
 - iv. Documentation
 - v. Publicity
 - vi. Maintenance
 - vii. Community-building
 - viii. Support
- b. What are impediments to doing more of this?
- c. What are the costs to us?
- d. What are the benefits to us?
- e. Are there things we can be doing proactively to make the eventual release to open source easier?

Additional Notes

All open source software is not equal. Key differences include the size, support network and licenses.

- Different types
 - Relatively small libraries and tools
 - Examples: Log4J, FlexiGrid
 - Tend to be selected and installed by developers
 - Back-end software libraries, middleware, operating systems, storage systems and tools
 - Tend to be hidden to end-users
 - Examples: Tomcat, Apache, JBOSS, Fedora, Linux, Heritrix, MySQL, Zend/PHP
 - Installed by the operations group, selected and used primarily by developers group in collaboration with operations
 - Larger applications with front-end components
 - Tend to be relatively turn-key solutions – easier to install and start using
 - Examples: DSpace, Archivist Toolkit, VuFind, BlackLight, EverGreen?, Drupal
- Different support network
 - Maintained by a single individual
 - Maintained by a group of individuals
 - Maintained by a single Institution
 - Maintained by a group of Institutions
- Different licenses
 - GPL, LGPL, BSD, MIT, Apache, etc., etc.
 - Free vs Open Source