IS THERE SUCH A THING AS FREE SOFTWARE?

THE PROS AND CONS OF USING OPEN SOURCE SOFTWARE AT UCLA

Thomas J. Trappler
UCLA Office of Information Technology

April 3, 2009
# Table of Contents

**Executive Summary** .................................................................................................................... 1

**Introduction** ............................................................................................................................... 3

**A Brief History of OSS** ............................................................................................................... 3

**OSS Defined** ................................................................................................................................ 4

**Licensing Issues** .......................................................................................................................... 5

**The Pros and Cons of Using OSS** .............................................................................................. 6

**Proposed Guidelines Regarding OSS Usage** ............................................................................ 7

  - Scenario 1: Researcher ................................................................................................................ 8
  - Scenario 2: UCLA-Wide Solution – Contribution Possible ........................................................... 8
  - Scenario 3: Department Solution – No Intent to Contribute ....................................................... 9
  - Scenario 4: End User .................................................................................................................... 10
  - Scenario 5: Community Source – Intent to Contribute ............................................................... 10

**Conclusion** .................................................................................................................................... 11

**Appendix A - The Open Source Definition** .............................................................................. 13

  - Introduction ................................................................................................................................ 13
  - 1. Free Redistribution .................................................................................................................. 13
  - 2. Source Code ............................................................................................................................. 13
  - 3. Derived Works .......................................................................................................................... 13
  - 4. Integrity of the Author's Source Code ...................................................................................... 13
  - 5. No Discrimination Against Persons or Groups ......................................................................... 14
  - 6. No Discrimination Against Fields of Endeavor ....................................................................... 14
  - 7. Distribution of License ............................................................................................................ 14
  - 8. License Must Not Be Specific To A Product ............................................................................. 14
  - 9. License Must Not Restrict Other Software ............................................................................. 14
  - 10. License Must Be Technology-Neutral .................................................................................... 14

**Appendix B - Software Attributes Comparison** ......................................................................... 15

**Appendix C - OSS Contributions** ............................................................................................ 22

  - Overview ..................................................................................................................................... 22
  - Contributor Agreements ................................................................................................................. 23
  - Reciprocal Licenses ....................................................................................................................... 24
  - Contribution Guidelines ................................................................................................................ 24
  - Code Review ................................................................................................................................. 25

**Appendix D - OSS Licenses** ........................................................................................................ 26

  - Common OSS License Types ....................................................................................................... 26
  - Key Concerns ............................................................................................................................... 27
APPENDIX E - OSS AS A VIABLE ALTERNATIVE

OVERVIEW

PROCUREMENT PRACTICES REVISIONS

APPENDIX F - OSS MATURITY EVALUATION GUIDELINES

GOALS

CRITERIA

MATURITY EVALUATION FRAMEWORKS

ATTACHMENT 1: CONTRIBUTION AGREEMENT FORMS

OPEN SOURCE SOFTWARE CONTRIBUTION AGREEMENT

OPEN SOURCE SOFTWARE CONTRIBUTION AGREEMENT - ONGOING

REVISION HISTORY

7/30/2008 Initial Draft

2/9/2009 Incorporated comments received from other UC campuses

IS THERE SUCH A THING AS FREE SOFTWARE?

EXECUTIVE SUMMARY
Each year UCLA spends approximately $9,000,000 on the purchase and maintenance of computer software. This software provides the foundation of UCLA’s information technology infrastructure. Today’s higher education environment is marked by heightened accountability and decreased budgets. In such an environment, no institution can afford to ignore alternative approaches that may result in more effective and less costly solutions.

Open Source Software (OSS) can serve as a viable alternative to the continued use of traditional Proprietary Software (PS) and should be equally evaluated as a potential solution based upon the individual needs of each project.

OSS has been used at UCLA on an ad hoc basis for many years, but the recent decision to use the OSS product Moodle as the basis for UCLA’s Common Collaboration and Learning Environment (CCLE) and the Shibboleth protocol for authentication are our first formal, enterprise-wide implementations of OSS solutions. UCLA’s expanding use of OSS is but one example of a larger trend in higher education, in which the use of OSS is expected to double over the next four years.

As with any software, the use of OSS comes with both pros and cons. To ensure that OSS is effectively evaluated, deployed and maintained at UCLA, we need guidelines and/or processes to:

- ensure that OSS products are considered alongside PS products during the acquisition process;
- ensure that only OSS products which meet an appropriate maturity level are selected for use;
- ensure that any contribution of UCLA developed code to an OSS project is in the best interests of UCLA; and
- identify any OSS licenses that should not be used.

We can no longer afford to limit ourselves to using traditional fee-based PS. The use of OSS under the parameters outlined above can provide UCLA with more effective and less costly software solutions. This document provides guidance to potential users of OSS through various scenarios and checklists.
Final clearance for contributions to an OSS project must be obtained from the Office of Intellectual Property and Industry-Sponsored Research. Current forms can be obtained from the OIP-ISR website. Their staff can also advise potential users and contributors on specific legal or licensing issues.
INTRODUCTION

Today’s higher education environment is marked by heightened accountability and decreased budgets. These trends are likely to continue, if not increase in scope. In such an environment, no institution can afford to ignore alternative approaches that may result in more effective and less costly solutions. Open Source Software (OSS) can serve as a viable alternative to the continued use of traditional Proprietary Software (PS) and should be equally evaluated as a potential solution based upon the individual needs of each project.

OSS has been used at UCLA on an ad hoc basis for many years, but the recent decision to use the OSS product Moodle as the basis for UCLA’s Common Collaboration and Learning Environment (CCLE) is our first formal, enterprise-wide implementation of an OSS solution. UCLA’s expanding use of OSS is but one example of a larger trend in higher education in which the use of OSS is expected to double over the next four years. “OSS is already a part of most higher education institutions and, if it is not, chances are high that it is part of an institution that stakeholders (such as researchers and students) collaborate with.”

As with any software, the use of OSS comes with both pros and cons. “As OSS becomes ubiquitous, Gartner expects to see a corresponding increase in problems arising from poor management controls in organizations that misunderstand OSS.” To maximize the pros and minimize the cons associated with this new tool, UCLA needs to develop processes by which it can ensure that OSS is effectively evaluated, deployed and maintained.

A BRIEF HISTORY OF OSS

OSS began in 1977 when three graduate students at UC Berkeley started distributing open source versions of UNIX under the Berkeley Software Distribution (BSD) license. Since that time, the use and influence of OSS has continued to grow. Much of the Internet was built on OSS in the 1970s, and much of the IT “cloud” is being built upon OSS today. Once considered to be an option fit only for bleeding-edge technology organizations, today OSS is increasingly being used for mission-critical infrastructure by conservative IT organizations that regard cost and risk mitigation as their primary concerns.

1 “Open Source Poised for Surge in Education” (Campus Technology), by David Nagel, 1/24/08
2 “Open Source in Higher Education, 2008” (Gartner), by Jan-Martin Lowendahl, 3/24/08
3 “Open Source in Contracts and Legal Issues, 2008” (Gartner), by Steward Buchanan and Jane B. Disbrow, 3/26/08
4 “Open Source in Higher Education” (Educause Quarterly, Number 2, 2008) by Shaheen E. Lakhan and Kavita Jhunjhunwala
The variety of available OSS product options is rapidly expanding. In 2005, SourceForge, the largest OSS development website, had over 74,000 OSS projects registered, today there are over 180,000.\(^5\) OSS has moved beyond its early phase which consisted primarily of IT infrastructure solutions to now include middle-ware and applications. While many of these products may not be enterprise class, the Burton Group finds that “for many applications there are compelling open source alternatives that should be considered.”\(^6\) The pedigree of available OSS products ranges from those developed by a loose-knit community of students working at night to improve their programming skills, to those developed by such well established companies as IBM, Novell and Sun using conventional software development methods.\(^7\)

Gartner predicts that “By 2012, more than 90% of enterprises will use open source in direct or embedded forms.”\(^8\) Governments around the world, including New Zealand, South Africa, the United Kingdom and California, have already recognized the benefits of OSS and have issued policies or guidelines promoting or mandating the evaluation and use of OSS.

A variation of OSS worthy of special note is Community Source Software (CSS). “Community source is a software development and deployment model that aims to coordinate the work of different user IT organizations sharing the same purpose and set of common requirements.”\(^9\) CSS is increasingly being used in higher education, often due to lack of viable PS alternatives. Prominent examples of CSS are Kuali, Sakai and uPortal.

**OSS DEFINED**

OSS is defined less by what the software does than what the user can do with the software. From a functional perspective, the results that can be achieved with OSS are similar to those that can be achieved with PS. What primarily distinguishes OSS from PS are its development and licensing models.

OSS is typically developed in an open, collaborative manner by a broad community of programmers with similar needs and/or goals regarding the desired features and

---

\(^5\) [http://sourceforge.net/index.php](http://sourceforge.net/index.php) These figures are intended to show a relative increase. Many of the OSS projects registered on SourceForge are single member supported, or no longer active.

\(^6\) “Open Source Software: Risks and Rewards” (Burton Group) by Gary Hein, 8/3/05

\(^7\) “Open Source Software briefing paper” (JISC) by Randy Metcalfe and Sebastian Rahtz, 1/17/06

\(^8\) “The State of Open Source, 2008” (Gartner) by Yefim V. Natis, George J. Weiss, Mark Driver, Nicholas Gall, Daniel Sholler, Brian Prentice, 4/3/08

\(^9\) “How Open Source Is Changing the Shape of IT” (presentation at 4/08 Gartner Symposium) by Mark Driver
functionality of the end product. These products are often developed either to provide an alternative to an existing PS product or to satisfy a need that no PS product currently fulfills. This contrasts with PS product development which is typically a closed process conducted by the programmer employees of an individual vendor whose priorities are based upon the financial goals of the vendor.

The primary ways in which an OSS license differs from a traditional PS license are simple, but of great significance:

- There are no license fees to use the software
- The software source code is available to all users

To understand OSS, one needs to understand the terms of OSS licenses. It is widely accepted that for a software license to be considered Open Source, it must have been approved by the Open Source Initiative (OSI). To-date, 72 different OSS licenses have been approved by the OSI\(^\text{10}\). OSI OSS license approval is based upon the license’s compliance with the terms of the OSI’s “Open Source Definition” (OSD)\(^\text{11}\) which includes requirements related to distribution, availability of source code, and eligibility to use. The complete text of the OSD can be found in Appendix A.

**LICENSING ISSUES**

Software is a form of intellectual property. When acquiring software, an organization typically does not acquire ownership of the underlying source code, what they acquire is a license to use the software. The terms of this license allow for use of the software within certain defined parameters. The legal and financial risks associated with license violation are similar for both OSS and PS, so UCLA should evaluate an OSS license as thoroughly as we would a license from any PS vendor.

While each OSS license has terms that fall within the OSD, the specific terms of each will vary, so it is important to clearly identify and become familiar with the details of the license for each specific OSS product under consideration. The primary areas of concern regarding OSS licenses include: the obligations being assumed by use of the product, how the product itself (including its source code) may be used, and the degree to which the terms of an OSS license are compatible with those of other software licenses.

While there are numerous variations of OSS licenses, they generally fall within one of the two most common categories: Academic and Reciprocal. Academic licenses (such as the

---

\(^{10}\) [http://www.opensource.org/licenses/alphabetical](http://www.opensource.org/licenses/alphabetical)

\(^{11}\) [http://www.opensource.org/docs/osd](http://www.opensource.org/docs/osd)
Berkeley Software Distribution (BSD)) tend to be more flexible for incorporating the OSS product into new works. Reciprocal licenses (such as the General Public License (GPL)) tend to be more restrictive regarding how the OSS product may be incorporated into new works, so particular attention should be paid to this type of license. More detailed information regarding OSS licenses can be found in Appendix D.

THE PROS AND CONS OF USING OSS

There are many differing opinions and misconceptions about what OSS is and the pros and cons associated with its use. The decision of whether or not to adopt an OSS solution should be based upon practical reasons, not philosophical opinions. OSS should be viewed simply as another licensing and business model to be considered alongside traditional PS products as a potential solution during any software acquisition process.

It is important to evaluate all potential software solutions (OSS or PS) on a case-by-case basis based upon the specific needs of each project. The attributes to be considered when evaluating OSS are essentially the same as those to be considered when evaluating PS and include cost, license terms, fitness of purpose and reliability. A chart listing key software attributes along with the pros and/or cons of both OSS and PS for each can be found in Appendix B.

Most organizations initially deploy OSS to save money because no license fee is required to use OSS products. The use of OSS can provide other benefits including improved flexibility for customization, reduced likelihood of vendor lock-in, and easier license compliance.

OSS is not a panacea, and along with its benefits come some risks, including the potential increased need for in-house support, and some complex intellectual property issues related to incorporating OSS in new works and contributing new code to existing OSS projects. More detailed information regarding the issues related to contributing code can be found in Appendix C.
PROPOSED GUIDELINES REGARDING OSS USAGE

NOTE – Subsequent to the initial release of this document, UCLA has developed and implemented an OSS contribution process. For all software code being contributed to an OSS project on behalf of UCLA, the employee that developed the code must now complete either a “one time” or “ongoing” contribution form. The forms are available on the Office of Intellectual Property and Industry Sponsored Research’s website at:

One-Time OSS Contribution Agreement

Ongoing OSS Contribution Agreement
http://www.research.ucla.edu/oipa/files/OpenSourceContribution-Ongoing_2-2-09.doc

UCLA needs to develop mechanisms by which it can maximize the pros and minimize the cons associated with the use of OSS. Based upon UCLA’s organizational culture, establishment of a committee tasked with this responsibility would be the most likely mechanism to be successful. The committee should be comprised of key stakeholders in this arena, including subject matter experts in the areas of intellectual property, legal affairs and IT licensing.

The committee should develop usage appropriate UCLA-wide guidelines to ensure that OSS usage is effectively managed. The guidelines to be established should provide for the following:

• Ensure that OSS products are reviewed alongside PS products as equally viable potential solutions to specific UCLA needs
• Ensure that OSS product maturity is effectively evaluated in relation to the individual needs of each specific project
• Ensure that any OSS contributions are made in accordance with UCLA’s best interest
• Identify any OSS licenses with terms that are not acceptable for use at UCLA.

To ensure that these guidelines effectively fulfill UCLA’s needs, they should be reviewed by, and incorporate feedback from, key UCLA stakeholders such as faculty, IT decision-makers and administrators who are responsible for related matters. Subsequent to this review, and prior to implementation, the guidelines should be vetted and approved via UCLA’s standard governance channels. To achieve maximum effectiveness, these guidelines should be designed to be straightforward and easy to follow to ensure that end user compliance is as easy as possible.

The following OSS usage scenarios are intended to serve as examples of common types of OSS usage at UCLA. The scenarios include: a researcher who intends to incorporate OSS code in a new technology being developed; UCLA use of an OSS product where contribution is not required, but could occur; department use of an OSS product where there is no intention to contribute; end user use of an OSS product where there is no likelihood of
contribution; and UCLA participation in a CSS project where UCLA has committed to contribute.

SCENARIO 1: RESEARCHER

In this scenario, a UCLA faculty member is conducting research to develop a new type of software. The faculty member wants to incorporate parts of an existing OSS product into the new software being developed.

Under this scenario, the faculty member is best positioned to ascertain whether or not the OSS product provides the features, functions and maturity level appropriate to their research.

The faculty member may not be as knowledgeable about the pertinent issues related to the OSS product’s license. Areas of concern to both the faculty member and UCLA include:

Is the faculty member’s research funded by an external agency? If so, are there any conflicts between the terms of the OSS license and the terms associated with the external funding? The issues will be different depending upon the source of the external funding (federal, private).

Is there an intention and/or requirement for the new software under development to be commercialized? If so, do the terms of the OSS product’s license allow for it to be incorporated into and sold as part of a proprietary product? For example, if OSS using a Reciprocal license is incorporated in the new software being developed, that could significantly limit the ability to commercialize that new software.

The need to establish UCLA mechanisms to mitigate issues related to this scenario is high. All OSS licenses should be carefully evaluated prior to being incorporated into new works developed at UCLA to ensure that the license terms are compatible with the goals and requirements of the research. UCLA resources should be identified and made available to conduct this type of evaluation on an ongoing basis.

SCENARIO 2: UCLA-WIDE SOLUTION – CONTRIBUTION POSSIBLE

In this scenario, the need for a UCLA-wide software solution is identified and an RFI/P/Q is developed in order to evaluate and determine the best software product to meet this need.

The best software product to meet UCLA’s needs could be OSS or PS. As part of the process, both should be equally considered and objectively evaluated in relation to the specific needs of the project. Current procurement practices were developed primarily to evaluate proprietary products. Since OSS is a relatively new development, current procurement practices may not be well suited to take into account the unique nature of OSS. Current
procurement practices should be reviewed, and revised as needed, to ensure that OSS solutions are objectively evaluated alongside PS solutions. More details regarding how to ensure that the procurement process provides an objective evaluation of OSS can be found in Appendix E.

Since the selected solution could have a UCLA-wide impact, it is important to ascertain whether or not the software being considered is sufficiently mature to meet UCLA’s needs. While OSS can provide a viable and competitive alternative to PS in many cases, all OSS products are not equal in terms of maturity, just as all PS products are not equal. It is important for UCLA to develop common guidelines by which the maturity of an OSS product can be evaluated. The overall goal of the guidelines is to facilitate the selection of an OSS product that meets UCLA’s requirements and is likely to have a long, useful lifespan. More detailed information regarding OSS maturity evaluation can be found in Appendix F.

Due to the long-term, UCLA-wide implications of selecting an OSS product to fulfill such a need, UCLA should carefully decide what level of involvement we will have with the community around that OSS product — Do we want to be an adopter only, or do we want to actively participate in the community? Active participation requires additional resources, but some key benefits of using an OSS product are derived from the other members of that product’s community. Failing to engage with the development and support community of an OSS product could detract from UCLA’s ability to receive the full benefit of that OSS product. One important way of participating in an OSS community is to contribute new code back to the community. It is important that such contributions be effectively managed. More detailed information regarding the issues related to OSS contributions can be found in Appendix C.

The need to establish UCLA mechanisms to mitigate issues related to this scenario is moderate to high. UCLA resources should be identified and dedicated to review, and revise as necessary, current procurement practices to ensure that OSS solutions are objectively evaluated alongside PS options. Additionally, UCLA resources should be identified and dedicated to establishing and disseminating guidelines in the areas of OSS product maturity and contributions. The availability of these guidelines will enable UCLA units to determine when OSS can be effectively used, and when it cannot.

See Scenario 5 for a CSS variation of Scenario 2.

**SCENARIO 3: DEPARTMENT SOLUTION – NO INTENT TO CONTRIBUTE**

In this scenario, a department decides to use a common and well established OSS product, such as Linux, as part of the department’s IT infrastructure.
Under this scenario, the department is best positioned to ascertain whether or not the OSS product provides the features, functions and maturity level appropriate to meet the department’s needs.

The department’s IT staff may possess a sufficient level of knowledge to develop customized code based upon the OSS product that could be contributed back to the OSS community. However, a contribution is unlikely to occur because the OSS product has been adopted solely for use as a solution to a specific and common department IT infrastructure need. No need or goal to customize the OSS product exists, and any benefit that could accrue to the department from contributing is minimal.

The need to establish UCLA mechanisms to mitigate issues related to this scenario is low. The guidelines regarding OSS maturity evaluation noted in Scenario 2 would be available for departmental reference if needed. No additional UCLA review or input mechanisms are warranted.

SCENARIO 4: END USER

In this scenario, a department Administrative Assistant (AA) unilaterally decides to download the OSS product Firefox to use as their primary Internet web browser. The act of downloading, installing and using Firefox automatically means that the AA has agreed to the terms of the Mozilla Public License which Firefox is licensed under.

The average AA generally does not have sufficient technical knowledge to develop customized code that could be contributed back to the Firefox community. Even if the AA were to do software programming as a hobby in their spare time, the fact that their job responsibilities are not IT related would make it clear that any potential contribution was on behalf of the AA as an individual as opposed to the AA as a representative of UCLA.

There is no need to establish UCLA mechanisms to mitigate issues related to this scenario. No UCLA review or input mechanisms are warranted.

SCENARIO 5: COMMUNITY SOURCE – INTENT TO CONTRIBUTE

UCLA is not currently participating in a CSS project. This scenario is only intended to cover the issues that may arise should UCLA decide to join a CSS project in the future.

In this scenario, UCLA determines that no viable PS solution exists to meet its administrative software needs. UCLA could develop and implement a wholly customized software solution, but there are high risks associated with such an undertaking. UCLA determines that a lower risk option to fulfill our needs is to become a participating member of the existing CSS Kuali.
By opting to participate in Kuali, we are sharing the burden and risks associated with such a large scale undertaking with the other participating universities that have similar needs. Additionally, we gain the benefit of being able to use the existing code base, and code subsequently developed by the other participating universities.

As a participating member, UCLA commits to assigning full-time programmer employees to develop a specific component of Kuali with the intention of contributing it back to the Kuali community.

Issues regarding objectively evaluating a CSS project, and its maturity level, as a solution to a UCLA-wide need would be similar to those outlined in Scenario 2. UCLA would want to evaluate the maturity level of Kuali as a viable solution for its needs prior to becoming a participating member. Detailed information regarding OSS maturity evaluation can be found in Appendix F.

The range of issues related to making contributions to a CSS project would be similar to those outlined in Scenario 2. It should be noted that under Scenario 2, contribution remains optional, while under this scenario, as a member of Kuali, UCLA is expected to actively contribute to the project. Detailed information regarding the issues related to OSS contributions can be found in Appendix C.

The need to establish UCLA mechanisms to mitigate issues related to this scenario is high, primarily due to the fact that UCLA as an institution will contribute a significant quantity of intellectual property to the community. In addition to the availability of established guidelines, UCLA resources should be identified and made available to review code and coordinate the contribution process on an ongoing basis.

Similar issues have already been faced at other UC campuses participating in CSS projects. A sample internal UC Contribution Agreement related to Sakai participation is incorporated as Attachment 1.

CONCLUSION
To maximize the pros and minimize the cons associated with the use of OSS, UCLA should develop a process by which the use of OSS can be effectively managed. This process should include the establishment of guidelines to: ensure that OSS products are considered alongside PS products during the acquisition process, ensure that only OSS products which meet an appropriate maturity level are selected for use, ensure that any contribution of UCLA developed code to an OSS project is in the best interests of UCLA, and identify any OSS licenses that should not be used.
APPENDIX A - THE OPEN SOURCE DEFINITION
*from http://www.opensource.org/docs/osd

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.
APPENDIX B - SOFTWARE ATTRIBUTES COMPARISON

*Note: The pros and cons of both OSS and PS can vary for each individual product. Linux is very different from a brand new OSS product, just as Microsoft Windows is very different from a PS start-up’s new product. For this reason, the information below includes some generalizations.

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>OPEN SOURCE SOFTWARE</th>
<th>PROPRIETARY SOFTWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Terms</td>
<td>OSS licenses are fairly concise and straightforward, making it easier to manage.</td>
<td>PS licenses are fairly lengthy, complex and largely oriented to the vendor’s benefit. Expert knowledge is required to review and negotiate revisions.</td>
</tr>
<tr>
<td></td>
<td>Some OSS licenses are not compatible with other OSS licenses. This can limit the ability to use some OSS products in concert with others.</td>
<td>License terms for PS vendor products can vary widely from those of other PS licenses.</td>
</tr>
<tr>
<td></td>
<td>The simplicity of OSS licenses makes it easier to ensure compliance. There is no requirement to track license usage in relation to licenses purchased.</td>
<td>The complexity of PS licenses can make compliance more difficult due to either usage tracking or lack of understanding of license terms.</td>
</tr>
<tr>
<td></td>
<td>There are currently over 70 different approved OSS licenses. The quantity of licenses and the differences among them can cause confusion.</td>
<td>Each PS vendor’s standard license terms are different from those of other PS vendors.</td>
</tr>
<tr>
<td>Intellectual Property (IP):</td>
<td>OSS licenses do not provide the user with IP indemnification. Indemnification may be available for purchase from vendors such as RedHat or SuSe.</td>
<td>The likelihood of patent infringement is the same whether you use OSS or PS, but it is possible to negotiate for a PS vendor to provide indemnification.</td>
</tr>
<tr>
<td></td>
<td>Reciprocal OSS licenses limit new works that incorporate an existing OSS product to being</td>
<td></td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>OPEN SOURCE SOFTWARE</td>
<td>PROPRIETARY SOFTWARE</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>re-distributed under the same license terms as the original OSS. Organizations (such as UCLA) that create IP need to thoroughly review license terms. Mixing source code from multiple products can lead to IP uncertainties. There is the possibility that OSS code could be inadvertently included in a new work that is subsequently distributed in a manner contrary to the OSS license.</td>
<td></td>
</tr>
<tr>
<td>Cost - License</td>
<td>No license fee for initial license acquisition, subsequent license quantity increases, license renewals, updates, upgrades, and/or home use.</td>
<td>Payment is required for initial license acquisition, subsequent license quantity increases, and upgrades. Payment may be required for license renewals, updates, and/or home use.</td>
</tr>
<tr>
<td>Cost - Support</td>
<td>External support is only available for a fee. External OSS support providers include: IBM, HP, Novell, Computer Associates and Red Hat. The open availability of source code introduces competitive market forces into software support. This generally leads to lower cost support.</td>
<td>Support is either included in the license cost such as with subscription licenses, or available for an additional annual fee such as with perpetual licenses. The lack of alternatives reduces user leverage during negotiations.</td>
</tr>
<tr>
<td></td>
<td>In-house support is easier. Since the source code is open, IT staff can learn how a product works by reading the actual code. The costs associated with any in-house support are also lower.</td>
<td>In-house support is possible, but is made more challenging due to lack of access to the source code.</td>
</tr>
</tbody>
</table>

12 “Open Source in Contracts and Legal Issues, 2008” (Gartner) by Steward Buchanan, Jane B. Disbrow, 3/26/08
<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>OPEN SOURCE SOFTWARE</th>
<th>PROPRIETARY SOFTWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>support staff should be identified to effectively compare to PS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support is also available via the user community. “The responsiveness and accuracy of support found on mailing lists was far superior to what is normally associated with commercial vendor support.”¹³</td>
<td></td>
</tr>
<tr>
<td>Cost - Other Issues</td>
<td>Additional internal resources may be needed in order to evaluate OSS product options and/or participate in the OSS community.</td>
<td>OSS is still less common than PS, so there are fewer PS products that are compatible with OSS. Choosing an OSS solution may prevent the ability to use some PS products. For example, there are numerous PS products that will work with Windows, but not Linux.</td>
</tr>
<tr>
<td></td>
<td>OSS can often run on older hardware more efficiently than PS.¹⁴</td>
<td></td>
</tr>
<tr>
<td>Fitness of Purpose: Does it provide the required features and functions?</td>
<td>Evaluation OSS licenses and regular OSS licenses are the same, so evaluation licenses are available at no cost and have no expiration date. Participation in the OSS product's community can enable an organization to influence the direction of the product in a way that aligns with their own needs.</td>
<td>Evaluation licenses often have an associated fee and a limited duration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A vendor can unilaterally change the direction of a product, or decide to discontinue it due to lack of profitability, merger/acquisition and/or bankruptcy.</td>
</tr>
</tbody>
</table>

¹³ “Open Source Software: Risks and Rewards” (Burton Group) by Gary Hein, 8/3/05
¹⁴ “Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!” (pg. 59, #4 & #5) by David A. Wheeler, 4/16/07, [www.dwheeler.com/oss_fs_why.html](http://www.dwheeler.com/oss_fs_why.html)
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Open Source Software</th>
<th>Proprietary Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is not always obvious how to identify and obtain a particular OSS solution. Additional internal resources may be required to do so.</td>
<td>PS vendors have marketing budgets, so there are often clearer avenues for identifying and obtaining products than with OSS.</td>
<td></td>
</tr>
<tr>
<td>Flexibility: How easy is it to customize the software to meet an organization's specific needs?</td>
<td>Since the source code is open, products can be customized in any way that meets one's needs, limited only by in-house resources or the ability to outsource this customization. The availability of the source code also makes it easier for others to customize. Often the entire OSS community can then benefit from such changes. OSS is commonly platform independent which provides greater flexibility and supports a wider developer and user audience. OSS usage provides more decision points, not just in software, but support and hardware as well. More choices = greater flexibility.</td>
<td>PS vendors do not generally allow users to alter or customize their products. In some cases customization may be achieved via additional payment to the vendor. Mature PS products still generally offer more features than the OSS counterparts.</td>
</tr>
<tr>
<td>Project Management: How well organized is the software development process?</td>
<td>There is the impression that OSS product development isn't well managed because no corporate entity is in charge. Larger, more mature OSS projects tend to be better managed. Burton Group notes that &quot;the OSS development</td>
<td>There is the impression that PS product development is more effectively managed because a corporate entity is responsible. This is not always true, particularly for smaller PS vendors.</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>OPEN SOURCE SOFTWARE</td>
<td>PROPRIETARY SOFTWARE</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Quality: How many errors are in the code?</td>
<td>There is the concern that since no vendor is responsible, no one is accountable for code quality. The opposite opinion is that OSS products “are often more reliable than products with closed source code because everything is continually peer-reviewed through use of the software.”[15]</td>
<td>A PS vendor can have numerous paid, full-time software developers working to ensure that code is error free. Nonetheless, it is rare that a PS software product never requires patches or bug fixes.</td>
</tr>
<tr>
<td></td>
<td>If support is in-house, IT staff will need to proactively maintain knowledge of updates and upgrades. Identifying a reliable source for version updates can be challenging without third party support.</td>
<td>PS vendors usually proactively provide notification of updates and upgrades. The PS vendor is usually a sole and trusted source of version updates for their product.</td>
</tr>
<tr>
<td>Reliability: How long will a product remain available, and stay in operation without user intervention?</td>
<td>OSS products require sustained community participation from skilled programmers to remain viable. While the focus of an OSS product may change direction, a user has the right to remain indefinitely on any version they choose. OSS products do not provide a warranty.</td>
<td>The continued availability of a PS product is based upon its commercial viability. PS licensing options such as annual licenses may not allow a user to remain on a prior version. PS vendors could provide a warranty, but most do not. Most PS licenses state that the software is sold on an</td>
</tr>
</tbody>
</table>

[15] “Open Source Software: Risks and Rewards” (Burton Group) by Gary Hein, 8/3/05
<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>OPEN SOURCE SOFTWARE</th>
<th>PROPRIETARY SOFTWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security: How resistant is the software to unauthorized actions (viruses, hacking, etc.)?</td>
<td>The code is open, so all users are able to view the code. More people viewing the code leads to problems being discovered and fixed more quickly.</td>
<td>A PS vendor can have numerous full-time software developers who are paid to ensure that code is secure.</td>
</tr>
<tr>
<td></td>
<td>Since PS code is not open, it is more difficult for hackers and malicious users to determine how to exploit potential vulnerabilities.</td>
<td>The code is not open, so it is more difficult for hackers and malicious users to determine how to exploit potential vulnerabilities.</td>
</tr>
<tr>
<td></td>
<td>“In the long run, openness of the source will increase its security…”18</td>
<td>“With closed source systems the perceived exposure may appear to be low, while the actual exposure… may be much higher.”19</td>
</tr>
<tr>
<td>Switching Costs: High transition costs are one of the primary reasons organizations remain with a current software solution.</td>
<td>The primary component of switching costs with OSS is the retraining of IT and end user staff.</td>
<td>In addition to retraining IT and end user staff, PS switching costs include the need to pay the license fee for the new product concurrent with the old product during transition.</td>
</tr>
</tbody>
</table>

---


18 Ibid

19 Ibid
<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>OPEN SOURCE SOFTWARE</th>
<th>PROPRIETARY SOFTWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-Friendliness: How easy is the software to use?</td>
<td>OSS’ origins in IT infrastructure have given it the image of having poor user interfaces “built by IT for IT”. As OSS product options expand to applications, this is changing. Examples include: Firefox, OpenOffice, KDE, and Ubuntu Linux.</td>
<td>User-friendliness can vary by product, but PS products generally have better user interfaces.</td>
</tr>
</tbody>
</table>
APPENDIX C - OSS CONTRIBUTIONS

NOTE – Subsequent to the initial release of this document, UCLA has developed and implemented an OSS contribution process. For all software code being contributed to an OSS project on behalf of UCLA, the employee that developed the code must now complete either a “one time” or “ongoing” contribution form. The forms are available on the Office of Intellectual Property and Industry Sponsored Research’s website at:

One-Time OSS Contribution Agreement

Ongoing OSS Contribution Agreement
http://www.research.ucla.edu/oipa/files/OpenSourceContribution-Ongoing_2-2-09.doc

OVERVIEW

“Good practice suggests that if you wish to avail yourself of [community] support, you should also be willing to provide support for other members of the community when you are able.”20 Simply put, it is easier to ask for and receive help from a community when one also gives back to that community. One key way of participating in an OSS community is by contributing new code. There are many sensitive aspects to making contributions, so the process needs to be effectively managed.

“Contribution is similar to purchasing a product in that resources are devoted, but rather than licensing and maintenance fees, staff time is exchanged.”21 Contributing code can provide other benefits beyond the generation of community goodwill. Contributing can allow UCLA to influence the direction of the OSS product to ensure that it continues to align with our goals. Selecting an OSS product is an important long term investment, and every effort that UCLA makes to contribute to that product helps to ensure its ongoing success and protect UCLA’s investment. Contributing could also be considered to align with UCLA’s mission to create and disseminate knowledge for the betterment of our global society22.

Code developed by UCLA for an OSS product we use is likely done to customize the product to meet our specific needs. Each time UCLA upgrades to a new version of that OSS product, we have to expend resources to develop the same code customization to apply to the new version. If UCLA contributes that code back to the OSS community for incorporation into

20 “Top Tips for Selecting Open Source Software” (OSS Watch)
http://www.oss-watch.ac.uk/resources/tips.xml

http://www.wasc.ucla.edu/cpr_endnotes/Misson_Statement.pdf
the core product for all subsequent versions, then UCLA will save resources by not having to
develop the same customized code for each new version. The Open University has stated
that “Ensuring that as many OU developments as possible are integrated into the core of
Moodle simplifies the task of merging code every time a new version of Moodle is
released.”

Submitted code contributions generally have to go through a review process before being
accepted. Just because code is contributed does not guarantee that it will be approved for
incorporation into the core product. It is important to ensure that any UCLA code
contributions meet a minimum level of quality and usefulness. Doing so can establish
UCLA’s reputation within the community as a useful contributor.

CONTRIBUTOR AGREEMENTS

When contributing to an OSS product, both the individual contributor and the organization
with which they are affiliated may be asked to sign a Contributor Agreement. Contributor
Agreements typically include:

- An acknowledgment of understanding of the terms under which the code is being
  contributed
- A permanent, non-exclusive license to the contributed code
- Representations that the contributor is the author of the code and/or has the right to
  contribute it
- A statement identifying any code authored by others that is included in the
  contributed code
- A disclaimer to protect the contributor from liability should the contributed code not
  function as expected
- In some cases, an assignment in ownership of the copyright to the contributed code.

One element of many Contributor Agreements that is of particular concern is the grant of
patent license. This grant is intended to provide assurances that the contributor has not
contributed code which infringes on their own patent. The language of some contributor
agreements suggests that contributors must provide a patent license to all patents that they
may have an interest in. Such Contributor Agreements do not appear to take into account
the possibility that a contributor’s patents may have already been licensed to a third party, or
have other commitments related to research sponsorship. Additionally, for some potential
contributor organizations, the process by which the relevant patents and agreements are
identified would require a significant amount of resources.

23 “Large-Scale Open Source E-Learning Systems at the Open University UK” (ECAR Research Bulletin) by Niall
Sclater, 6/10/08
Contributor Agreements should be vetted by UCLA to ensure that the terms of contribution are acceptable. Guidelines regarding acceptable and unacceptable Contributor Agreements should be established and made available to UCLA units.

**RECIPROCAL LICENSES**

The areas of concern regarding contributions primarily relate to OSS products with Reciprocal licenses. Since the predominant form of Reciprocal license is the GPL, UCLA should develop a balanced position regarding GPL contributions. Prohibiting the use of OSS products licensed under the GPL is unlikely to be helpful or effective due to the quantity of useful OSS products licensed under the GPL, and the fact that one of them is Moodle. UCLA could attempt to prohibit contributions to GPL licensed OSS, but this would result in the loss of most, if not all, of the benefits identified above. The best course of action is for UCLA to manage the process of making contributions to GPL OSS products by providing effective guidelines regarding the circumstances under which contributions may be made.

**CONTRIBUTION GUIDELINES**

UCLA’s contribution guidelines should take into account:

- Who is making the contribution (staff, faculty, contractor/consultant, etc.). The pertinent laws and policies may vary for each. It may be necessary to obtain a statement of understanding of the contribution terms from the individual who developed the code to be contributed prior to its contribution. See Attachment 1 for a sample statement.
- UCLA’s internal intellectual property and/or royalty-sharing policies and how they relate such contributions. Some policies in these areas may not directly address non-commercial uses of intellectual property.
- A process by which it can be ascertained that the code being contributed was fully created by UCLA and/or its representatives, and does not contain the intellectual property of others.
- A process by which it can be ascertained that the code being contributed does not have any prior conflicting license obligations. Code developed under some form of externally sponsored research should be closely reviewed for this.
- A process by which to ascertain whether or not the code being contributed is intellectual property that UCLA would like to retain the exclusive rights to. If the contribution is new code for a product licensed under the GPL, then UCLA could choose to use the code solely for internal purposes, but we would not have the right externally distribute it for a profit.

It may be useful for the process mentioned immediately above to include a mechanism by which to determine the potential commercial value of the new work.
The efforts undertaken to retain exclusive rights should be appropriate relative to the potential commercial value of the new work.

In cases where an OSS solution has already been vetted through the appropriate process and identified as a UCLA-wide solution, then the contribution of UCLA developed code to that OSS project should be understood to be in the best interest of UCLA. Examples of this would be contributions to Moodle or Shibboleth (which serves as the foundation for IAMUCLA).

**CODE REVIEW**

There are some software products available that serve to determine what, if any, previously developed code is incorporated into any given software product. The leading software solutions with such functionality are Black Duck, Palamida and OpenLogic (includes free open source tools/versions – OSS Discovery and OpenLogic Exchange). None of these products can guarantee to catch all previous code in all instances, and effective implementations can be costly. A low tech, lower cost, and relatively effective mechanism by which to obtain much of the information to answer the questions above is to check with the individual who initially developed the code to be contributed. The University of California has already developed its own individual Contribution Agreement form (see Attachment 1) which could be leveraged for this purpose.
APPENDIX D - OSS LICENSES

It is important to note that the ease with which OSS can be acquired (via free download), and the fact that the license terms are agreed to by the act of downloading, installing and/or using the product, make it challenging to manage OSS usage and related license obligations. This challenge is similar to that presented by PS click-through and shrink-wrap licenses. Many courts world-wide agree that clicking on “I ACCEPT” or tearing the shrink-wrap is sufficient evidence of the licensee accepting the license terms.24

COMMON OSS LICENSE TYPES

While there are numerous variations of OSS licenses, they generally fall within one of the two most common categories:

**Academic**: This type of OSS license was initially created by academic institutions. The most common example is the Berkeley Software Distribution (BSD) license developed at UC Berkeley. This type of license allows the software to be used for any purpose with no obligation on the part of the licensee to distribute the source code with derivative works. Anyone can use the software for any purpose – including for creating proprietary collective and derivative works – without having to contribute anything back. An example of this would be Apple using BSD licensed software as the basis for commercializing their computer operating system software.

**Reciprocal**: This type of license was developed subsequent to the BSD. The most common example is the General Public License (GPL) which was developed by the GNU project founded in 1984. This type of license allows software to be used for most purposes, but requires that creators of derivative works distribute those works under the same license. This requirement is intended to keep any software based upon an OSS product freely available, the same as the original product was, and prevent it from being incorporated into a closed source product. A large number of OSS products are released under the GPL, so it is crucial that its implications be addressed. A prominent example of an OSS product licensed under the GPL is Moodle, the foundation of UCLA’s campus-wide CCLE. Another widely used example GPL licensed software is Linux.

The license implications for OSS products used solely for internal purposes are generally fairly minor. More significant licensing concerns arise when an organization creates works derived from an OSS product and either contributes the code back to the OSS product or wants to re-distribute those works outside of their organization. The appropriate level of license review should be based upon the intended type of use.

24 “Open Source Licensing” by Lawrence Rosen
KEY CONCERNS

Key areas of OSS licensing concerns include:

**Obligations:** What obligations are being assumed by the user of the software? Do those obligations then accrue to UCLA? Does that user have the right to make such a commitment on behalf of UCLA? All of these are important to understanding what actions UCLA will need to take to ensure appropriate compliance with the obligations incurred.

**Inbound/Outbound:** OSS licenses generally include terms regarding both “inbound” and “outbound” use of the product. “Inbound” describes how an organization can use the OSS for its own internal purposes. “Outbound” describes how an organization can use OSS within intellectual property that it creates and how those works can be re-distributed outside of the organization. Since UCLA generates a significant amount of intellectual property, understanding the implications of “outbound” license terms is of key importance.

**Compatibility:** Are the license terms of the OSS product compatible with those of other product licenses (OSS or PS) that may be used in concert? If a project includes multiple OSS products under several different OSS licenses, it is possible that the licenses may be incompatible. For example, the BSD is compatible w/GPL, but the converse is not true because the GPL requires that derivative works be licensed as a whole at no charge to third parties under this license.
APPENDIX E - OSS AS A VIABLE ALTERNATIVE

OVERVIEW

The California Performance Review initiated by Governor Schwarzenegger recommends that “the state should more extensively consider the use of open source software, which can in many cases provide the same functionality as closed source software at a much lower total cost of ownership” and that state agencies should seek to “incorporate open source software as a viable alternative to any software procurement.”25 The most opportune time to consider an OSS solution is either when there is no current system in place, or when the current PS or custom developed solution reaches the end of its useful life.

Traditionally, PS vendors employ sales staff to respond to tenders and answer questions regarding their products. OSS product communities are not usually structured to be able to provide such services. This can lead to viable OSS solutions being overlooked during an evaluation process.

Tactically, simply including OSS in a procurement evaluation process can have benefits even if the OSS product is not determined to be the best solution. Including evaluation of OSS products/companies in the procurement decision process expands competition and increases the pressure on PS vendors to reduce costs, increase performance and better tailor their products to meet our needs.

PROCUREMENT PRACTICES REVISIONS

UCLA should develop a process by which OSS is equally evaluated during any related RFI/RFQ/RFP process. To achieve this, current procurement policies and practices may need to be revised. Key goals to be achieved by the resulting new policies and practices include:

Per the California Performance Review: “Since open source code solutions do not fit the traditional procurement model in that there is not usually a vendor promoting and proposing the product, it is recommended that state departments actively research and evaluate open source code alternatives prior to considering use of the traditional procurement model for software.”26

Ensure that RFI/P/Q questions are neutral and unbiased, and don’t inadvertently disadvantage OSS products. One example of an issue to review is --- How do company

26 Ibid
stability/viability questions asked as part of the current process take into account that OSS is developed by a community instead of by a corporate entity? In cases such as this, an OSS product maturity evaluation (see Appendix F) could be used as an equivalent to company stability.

Ensure that appropriate bids or bid equivalents are received or included in response to RFI/P/Qs. PS vendors are able to justify the substantial expenditures of responding to an RFI/P/Q because they have assumed those costs into their license fee structure. There is generally no mechanism by which an external response regarding a potential OSS solution can be received. UCLA may have to assign its own internal resources to identify relevant OSS solutions and conduct technical evaluations in lieu of an RFI/P/Q response. Once identified, it may be possible to get a representative from a specific OSS community or CSS foundation to come meet with us to discuss their project.

If external software support is required, the process should be structured so that third party OSS support vendors are proactively identified and notified so that they can potentially respond.

When the intention is to handle support in-house, the costs of the resources required to provide that in-house support should be identified and included in the evaluation to ensure an accurate cost comparison.
APPENDIX F - OSS MATURITY EVALUATION GUIDELINES

GOALS

OSS product maturity guidelines should accomplish the following:

- Enable maturity evaluations to be conducted on a case-by-case basis based upon each specific project’s needs.
- Identify required minimum rating thresholds for specific individual evaluation criteria. If a product does not meet these minimums, then it should be eliminated from consideration. This should occur as an initial review prior to conducting a full evaluation. If minimum requirements are not met, there is no need to conduct the full evaluation.
- Determine whether or not an OSS product’s overall maturity rating meets a pre-established minimum rating threshold. The minimum rating threshold identified should align with UCLA’s self-identified technology adoption profile --- are we an early, mainstream or late technology adopter?
- Minimum ratings should also take into account how an OSS product will be deployed. Deployment as a mission-critical application will require a higher maturity level than an exploratory project.
- Weighting for each rating criteria should be adjustable depending upon the specific project’s needs.
- Ratings methodology should be objective and consistent.
- Clearly define the reasoning behind each criteria and rating.
- Define an exception process to override when needed.
- Serve as an evolving document that changes and matures over time as elements of UCLA’s needs and OSS solutions evolve.

Gartner recommends avoiding “front-loaded efforts that focus too heavily on complete end-to-end solutions, but instead becomes mired in bureaucratic efforts that yield little real results.” 27

27 “How To Manage Open Source Software in the Enterprise” (presentation at 4/08 Gartner Symposium) by MarkDriver
CRITERIA

The OSS maturity evaluation guidelines should include ratings for the following criteria:

How long has the product been in existence?

For UCLA-wide solutions it will generally most desirable for the product to have been in existence long enough to have achieved a quantity and quality of community support sufficient to indicate long term viability.

UCLA should consider establishing a minimum time that an OSS product has been in existence based upon the specific needs of each project.

When was the last major version release? How often are updates/upgrades released?

The current version release number of an OSS product can be an indicator of maturity. A product that has reached its 1.0 release is often considered ready for enterprise use. Release version alone is not a sufficient indication of maturity as release levels may be arbitrarily applied.

The level of frequency of updates and upgrades can be good evidence of ongoing effort to develop and maintain the OSS product. Active projects tend to update “early and often”.

How extensively is the product used?

Is the usage rate stable, growing or declining?

The level and variety of community activity on the website, email lists and other support channels can provide good evidence regarding level of usage.

Is the product known to already be in use at an institution of comparable size and complexity to UCLA?

It can be useful to discuss with peers known to be using the product under consideration.

What are the goals and values of the OSS product and its community?

It is important to get an idea of the future direction of the product to ensure that it aligns with the UCLA’s desired direction.

It is also important to get an idea of the culture of the community around the OSS product to determine whether or not it is a viable match for UCLA’s culture.

OSS projects that embrace the core tenets of open source are likely to find the largest audience which provides the best chance at continued viability and growth.
What is the cost to setup the product?

Is a well-defined, trustworthy channel of acquisition available?

What level of in-house expertise exists? What is the cost of those in-house resources? If no in-house resources exist, then how easily can such resources be acquired (training, hiring) and at what cost?

What quality of community support is available?

It can be useful to review and monitor the OSS product’s knowledgebase and community support channels to obtain an understanding of the time it takes to receive a response, as well as the accuracy and thoroughness of the responses provided.

This can be further tested by posting a message to the list. If a reasonably prompt reply is received, this can be evidence of an active community ready to help.

Are there developers who are paid to contribute?

For some uses, depending upon community support may be sufficient. For mission-critical use, it will be desirable to have commercial support options.

Is third-party commercial technical support available?

The availability of commercial technical support can be a good indicator of the product’s overall level of sustainability. If there are enough organizations using the product for vendors to make a viable living from providing support, then that can be an indication of a large volume of users.

The cost of in-house support should be analyzed in comparison to commercial support.

What quantity/quality of documentation is available?

Is any documentation available? If so, is it well organized and easy to use? When was it last updated? How often is it updated?

Good documentation should enable you to trace a clear history of bug fixes and feature changes.

It is generally a good indication of an established volume of users if one or more books regarding the use of the software have recently been published.

How well designed is the software? What is the modularity level of the product?

This information can help identify the level of effort that would be necessary to adapt and/or extend the product to meet UCLA’s specific needs.
How many platforms are supported?

Does the product work with the necessary platforms (Windows, Macintosh, Linux, etc.)?

Is the product built upon open standards? How well does the product interoperate with other products?

Products not built upon open standards can lead to product lock-in which can lead to increased costs. UCLA’s data could be created and stored in a closed standard meaning it could become inaccessible should the OSS product fail to remain viable.

Lack of open standards can increase the cost to migrate to another solution.

Open standards can indicate a greater likelihood of the ability to effectively integrate with other products.

MATURITY EVALUATION FRAMEWORKS

A number of OSS maturity evaluation frameworks exist, including the Open Source Maturity Model (OSMM) from Navica, the Business Readiness Rating Model (BRR), the Qualification and Selection of Open Source Software (QSOS), and the OS Maturity Model from CapGemini. While the ratings scale for each of these frameworks varies, they tend to cover evaluation criteria similar to those noted above. UCLA should consider adopting or adapting one of these existing frameworks as the foundation of its OSS maturity evaluation guidelines. Burton Group recommends the OSMM from Navica as the best choice.28 OSMM provides a good balance between thoroughness and ease of use.

28 “Assessing Open Source Software Projects” (Burton Group) by Richard Monson-Haefel, 10/14/05
ATTACHMENT 1: CONTRIBUTION AGREEMENT FORMS

This attachment provides a sample copy of the forms required to be filed for all software code being contributed to an OSS project on behalf of UCLA. The current forms are available on the website of the Office of Intellectual Property and Industry Sponsored Research’s:

One-Time OSS Contribution Agreement

Ongoing OSS Contribution Agreement
http://www.research.ucla.edu/oipa/files/OpenSourceContribution-Ongoing_2-2-09.doc

The forms reproduced here are samples as of the publication date of this document. It is anticipated that these forms will be revised as practical experience is gained with contributions to open source initiatives. Therefore, contributors should download forms that are current at the time of submission from the OIP-ISR website.
OPEN SOURCE SOFTWARE CONTRIBUTION AGREEMENT

This form must be completed for each contribution of code to an existing Open Source Software (OSS) project that is being made on behalf of the University of California, Los Angeles (UCLA). Each UCLA employee that participated in the development of the code to be contributed should complete this form and email it to the UCLA Office of Intellectual Property at copyrights@research.ucla.edu.

1. For ease of reference, please provide a short name/title for the code to be contributed.

________________________________________________________________________

2. Name (in alphabetical order), UCLA Department, Phone Number, and Email Address of each UCLA employee who contributed to the development of this code. Each of the contributors listed below must separately complete and submit a copy this form.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

3. Describe the functionality of the code to be contributed.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
4. If the code to be contributed is related to an approved UCLA campus-wide information technology project, please identify that project by checking one of the boxes below.

☐ Common Collaboration and Learning Environment (CCLE)/Moodle
☐ IAMUCLA/Shibboleth
☐ Other (specify here) __________________________________________________

5. If the code to be contributed is not related to an approved UCLA campus-wide information technology project, then describe how the contribution of this code would benefit UCLA.
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

6. Does the code to be contributed only contain intellectual property that is either original to the developers listed in response to question #2 above or intellectual property that is already part of the OSS project being contributed to?

☐ Yes
☐ No. Describe which portions of the code to be contributed are not original to the developers listed above and/or are not already part of the OSS project being contributed to. Specify the source of each of these portions of the code.
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
7. What is the nature of the code to be contributed?

☐ Patch Existing Code/Bug Fix
☐ Enhancement
☐ “Obvious” New Functionality. This would be an easily recognized, readily apparent functionality that was previously lacking or missing from the software.
☐ “Novel” New Functionality. This would be a functionality, that has not been used before in this or any other application, that represents a new way of solving a problem or addressing a functional or technical requirement. (If you have selected this box, please complete an invention disclosure form prior contributing your code to the OSS project. Invention disclosure forms can be found at: http://www.research.ucla.edu/oipa/faculty.htm).

8. Code to be contributed must have already successfully passed an internal UCLA technical review. In the space below, provide the name of the entity that conducted the internal UCLA technical review, and the date on which the review was conducted.

________________________________________________________________________
________________________________________________________________________
I hereby declare that the foregoing is true and correct, and that I have read, understand and accept the following:

1) I understand that I am under no obligation to contribute this code and I agree to do so voluntarily.

2) I have read and understand the terms and conditions of the Contributor Agreement associated with the OSS project that the code is to be contributed to.

3) I acknowledge that I have an obligation to promptly disclose my patentable inventions to the University and to assign such inventions to the University.

4) I agree that all copyrighted works created by me in the performance of my employment by the University shall be considered Institutional Works, as described in the University’s Policy on Copyright Ownership (http://www.ucop.edu/ucophome/coordrev/policy/8-19-92att.html), and that such works are owned by the University. I will execute such documents as are necessary to assign all copyrights in such works to the University or to confirm the University’s ownership of copyright.

5) I agree to promptly disclose to the University software, code and other intellectual property developed by me in the performance of my employment by the University.

6) I have disclosed to the University patentable inventions and copyrighted works for which I am the inventor or author, that were made or created during the term of my employment or through the use of University resources or facilities, and that are necessary to exploit the inventions or software that are reasonably anticipated to be made or created under the project.

7) I have not knowingly incorporated into the materials that I developed for this contribution any software, code or other intellectual property that is not original to the developers listed above or already part of the OSS project being contributed to.

8) I consent to allow the University to grant to the OSS project that the code is being contributed to a no-cost copyright and/or patent license to the code being contributed under the terms of the OSS project’s Contribution Agreement. I understand that the grant of these licenses will preclude me from sharing in license revenue to which I would otherwise be entitled under the University’s Patent Policy and/or Copyright Policy.

9) The funding used for this project was not restricted and was appropriate for the work performed.

Signature______________________________________
Printed Name:__________________________________
Title:__________________________________________
Address:_______________________________________
Date: _________________________________________

Email this completed form to the UCLA Office of Intellectual Property at copyrights@research.ucla.edu.
OPEN SOURCE SOFTWARE CONTRIBUTION AGREEMENT - ONGOING

This form must be completed by each University of California, Los Angeles (UCLA) employee making ongoing contributions to the development of the code of an existing Open Source Software (OSS) project as part of their UCLA job responsibilities. Once completed, email this form to the UCLA Office of Intellectual Property at copyrights@research.ucla.edu.

1. Name, UCLA Department, Phone Number, and Email Address of the UCLA employee making ongoing contributions to the development of the code of an existing OSS project.

NAME:__________________________________________________________________
DEPARTMENT:___________________________________________________________
PHONE:_________________________________________________________________
EMAIL:_________________________________________________________________

2. If these ongoing code development contributions are related to an approved UCLA campus-wide information technology project, please identify that project by checking one of the boxes below.

☐ Common Collaboration and Learning Environment (CCLE)/Moodle
☐ IAMUCLA/Shibboleth
☐ Other (specify here) __________________________________________________

3. If these ongoing code development contributions are not related to an approved UCLA campus-wide information technology project, then describe how these code development contributions would benefit UCLA.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
I hereby declare that the foregoing is true and correct, and that I have read, understand and accept the following:

1) I understand that I am under no obligation to contribute this code and I agree to do so voluntarily.

2) I have read and understand the terms and conditions of the Contributor Agreement associated with the OSS project that the code is to be contributed to.

3) I acknowledge that I have an obligation to promptly disclose my patentable inventions to the University and to assign such inventions to the University.

4) I agree that all copyrighted works created by me in the performance of my employment by the University shall be considered Institutional Works, as described in the University’s Policy on Copyright Ownership (http://www.ucop.edu/ucophome/coordrev/policy/8-19-92att.html), and that such works are owned by the University. I will execute such documents as are necessary to assign all copyrights in such works to the University or to confirm the University’s ownership of copyright.

5) I agree to promptly disclose to the University software, code and other intellectual property developed by me in the performance of my employment by the University.

6) I have disclosed to the University patentable inventions and copyrighted works for which I am the inventor or author, that were made or created during the term of my employment or through the use of University resources or facilities, and that are necessary to exploit the inventions or software that are reasonably anticipated to be made or created under the project.

7) I will not knowingly incorporate into the materials that I develop for this OSS project any software, code or other intellectual property that is not original to the myself or already part of the OSS project being contributed to.

8) I consent to allow the University to grant to the OSS project a no-cost copyright and/or patent license to the code being developed under the terms of the OSS project’s Contribution Agreement. I understand that the grant of these licenses will preclude me from sharing in license revenue to which I would otherwise be entitled under the University’s Patent Policy and/or Copyright Policy.

9) The funding used for this project was not restricted and was appropriate for the work performed.

__________________________________________
Signature

__________________________________________
Printed Name:

__________________________________________
Title:

__________________________________________
Address:

__________________________________________
Date:

Email this completed form to the UCLA Office of Intellectual Property at copyrights@research.ucla.edu.