

Assessing Open Source Software Implementation Options from a Digital Preservation Perspective

Sri Vinay and Robert R. Downs

sri@ciesin.columbia.edu, rdowns@ciesin.columbia.edu

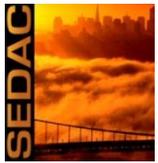
NASA Socioeconomic Data and Applications Center (SEDAC)
Center for International Earth Science Information Network
(CIESIN)

The Earth Institute, Columbia University

Open Geospatial Consortium
Technical Committee and Planning Committee (TC/PC) Meeting
Austin, Texas
March 20, 2012



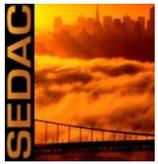
Software Dependencies in Digital Preservation



- Digital Preservation Systems
 - Content Management Systems, Databases
 - Digital repositories
 - Digital libraries
 - Digital asset management systems
 - Backup, replication, and recovery systems
- Digital Object Processing Functions
 - Ingest and assessment software
 - Integrity checking and virus inspection applications
 - Data conversion and integration (GIS, stat packages)
 - Transfer
- Digital Object Rendering and Use
 - Viewing and displaying content
 - Analysis and manipulation (GIS, stat packages)
 - Content modification



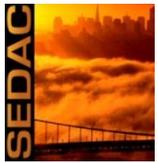
Software: a Digital Preservation Concern



- Digital objects, including scientific data, may be dependent upon software for rendering, either within the platform or externally
 - Enabling the future use of software contributes to digital preservation
- Some software may not include the original code, limiting its use on future platforms to those that are compatible
- Data loaded to some software platforms may become the vendor's property
- Some software licenses may not authorize:
 - Use on multiple devices
 - Use by another user
 - Use within another organization
 - Use after purchased licensing period
 - Porting code to another operating system
 - and more (read the fine print)



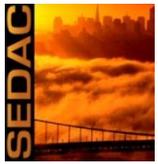
Open Source Software for Digital Preservation



- Open licenses, often without restrictions
 - Generally, not limited to purposes, locations, users, or organizations
 - Licenses vary (read the licenses)
- Modification opportunities
 - May be customized by internal developers
 - Community reviewed for various purposes
 - Being improved by external developers



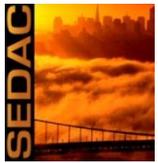
OS Software Alternatives



- Open Source Software (OSS)
- Vendor-Supported OSS
- Commercial Software (with or without OSS components)
- Combinations of these options



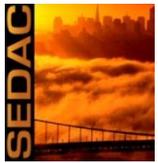
Adopting Open Source Software (OSS)



- Develop and share software as open source
 - License as open source to enable adoption and invite enhancements (Example OS licenses: Apache, BSD, MIT)
 - All OS licenses are not the same
 - Provide access within open source community (Apache, SourceForge, GitHub, etc.)
- Collaboratively contribute to OSS projects
 - Adopt OSS developed by others
 - Test, enhance, document, etc., to improve software



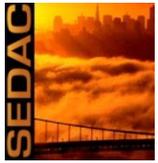
Adopting Vendor-Supported OSS



- Adopt OSS product provided by vendor
 - Vendor provides services such as installation, hosting, enhancement, customization, documentation, 24x7 help
- Vendor contributes enhancements to OSS
 - Enhancements meeting customer needs are also applicable to the OSS community



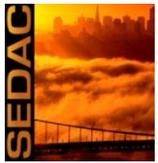
Adopting Commercial Software



- May or may not contain OSS components
 - Proprietary code may contain restrictions
 - Commercial and OSS license interactions between may be complex
- Identify common needs for enhancements among customers
 - Recommend enhancements to be released by vendor
- Collaborate on customizations within customer community
 - Develop modules for release as OSS



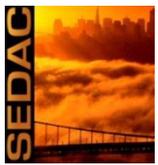
Adopting combinations of OSS and Commercial Products and Services



- System composed of commercial and OSS
 - Commercial and OSS products
 - Support from OSS community and from vendors
- Collaborating with OSS and customer communities
 - Developing OSS that integrates vendor products
 - Licensing vendor products that integrate OSS



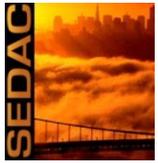
Assessing OSS for Digital Preservation



Software	Risks	Benefits	Preservation Options
Open Source Software (OSS)	OS community stagnation	Development & support from OS community and internal staff	Preserve, modify, use, and distribute in accordance with licenses
Vendor-Supported OSS	Change in vendor services	Development & support from vendor & OS community	Dependencies and restrictions may need to be identified
Commercial Software (with or without OSS components)	Product discontinuation, license limitations	Development & support from vendor & customer community	Additional licenses may be required to obtain rights for commercial components
Combinations of Commercial & OSS	OS community stagnation, change in vendor services or products, rights	Development & support from diversified communities	Identify interfaces between COTS and OSS; license or replace COTS



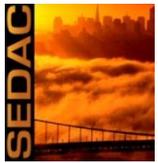
Using the Reuse Readiness Levels (RRLs)



- Assessing the reusability of OSS can indicate its potential for future use
- RRLs (2010) Topic Areas can be used to assess software reusability
 - Documentation
 - Extensibility
 - Intellectual Property Issues
 - Modularity
 - Packaging
 - Portability
 - Standards Compliance
 - Support
 - Verification and Testing
- Levels (1-9) can be used to assess the reusability of OSS for a given Topic Area
- Levels range from 1 (Limited reusability) to 9 (Demonstrated extensive reusability)
- Individual topic areas can be weighted to produce a score for an OSS component



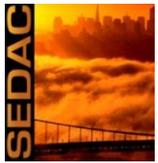
Assessing Reusability of OSS with RRLs



- RRL 1 – Limited reusability
 - the software is not recommended for reuse.
- RRL 2 – Initial reusability
 - software reuse is not practical.
- RRL 3 – Basic reusability
 - the software might be reusable by skilled users at substantial effort, cost, and risk.
- RRL 4 – Reuse is possible
 - the software might be reused by most users with some effort, cost, and risk.
- RRL 5 – Reuse is practical
 - the software could be reused by most users with reasonable cost and risk.
- RRL 6 – Software is reusable
 - software can be reused by most users although there may be some cost and risk.
- RRL 7 – Software is highly reusable
 - the software can be reused by most users with minimum cost and risk.
- RRL 8 – Demonstrated local reusability
 - the software has been reused by multiple users.
- RRL 9 – Demonstrated extensive reusability
 - the software is being reused by many classes of users over a wide range of systems



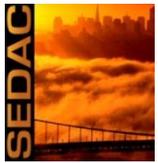
Example from RRLs: Intellectual Property Issues



- The legal rights for obtaining, using, modifying and distributing the asset.
 - A formal and documented explanation of the involved parties and roles, with binding statements describing any licensing mechanisms, ownership rights, restrictions, and user/consumer responsibilities related to the distribution and reuse of assets. The legal rights are established in accordance with the policies and laws of the organization that originally produced the software.
 - Potential adopters need to understand the intellectual property issues to know whether they have the authority to reuse the software.
- **Level 9** – Statements describing unrestricted rights, recommended citation, and developers embedded into product.
 - Multiple statements are embedded into the product describing unrestricted rights and any conditions for reuse, including commercial reuse, and the recommended citation. The list of developers is embedded in the source code of the product, in the documentation, and in the expression of the software upon execution. The intellectual property rights statements are expressed in legal language, machine-readable code, and in concise statements in language that can be understood by laypersons, such as a pre-written, recognizable license.



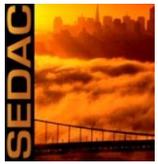
OSS for Digital Preservation: Summary



- Digital objects may require software to enable future use
 - To find, access, and analyze data and other digital objects
 - Migrate data from commercial to OS formats
- Assessing reusability of software to be preserved
 - Identify software needs for future levels of services
 - Identify criteria, such as Reuse Readiness Levels (RRLs), as thresholds of acceptability for digital preservation
- All OS licenses are not the same
 - OS licenses may contain incompatibilities
 - Dependencies may exist between commercial and OS licenses
 - Read commercial and OS licenses



References



Apache License, Version 2.0: Licensing. Available from the Open Source Initiative:
<http://www.opensource.org/licenses/apache2.0.php>

BSD License. Available from the Open Source Initiative:
<http://www.opensource.org/licenses/bsd-license.php>

Downs, R. R. 2012. Alternative Approaches to Collaborative Software Deployment Leveraging Open Source Software. Winter 2012 ESIP Federation Meeting, Washington, DC, January 4, 2012.

Downs, R. R., and J. Marshall. 2010. A proposal on using reuse readiness levels to measure software reusability. *Data Science Journal*, 9, 73-92. Available online at
<http://dx.doi.org/10.2481/dsj.009-007>

MIT License. Available from the Open Source Initiative:
<http://www.opensource.org/licenses/MIT>

Reuse Readiness Levels (RRLs) Version 1.0. 2010. NASA Earth Science Data Systems Software Reuse Working Group. Available online at
http://earthdata.nasa.gov/sites/default/files/esdswg/reuse/Resources/rrls/RRLs_v1.0.pdf

Yetman, G. and R. R. Downs. 2011. Models for Deploying Open Source and Commercial Software to Support Earth Science Data Processing and Distribution. IN23B-1452, 2011 Fall American Geophysical Union (AGU) Meeting, December 6, 2011, San Francisco, CA.