

Sustainable Software for the Future of Science: A User's Perspective

Robert R. Downs

rdowns@ciesin.columbia.edu

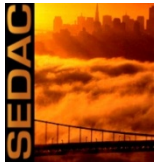
NASA Socioeconomic Data and Applications Center (SEDAC)
Center for International Earth Science Information Network
(CIESIN)
The Earth Institute, Columbia University

2014 SUMMER ESIP FEDERATION MEETING
Frisco, Colorado, 8-11 July 2014

Plenary Session
Wednesday 9 July 2014



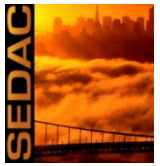
Software is Used throughout Research Lifecycle



- Science practice is improving to create new knowledge
 - Learning is influenced by methods, data, and results
 - Science continues to build on all aspects of previous research
- Scientific tools often include software components
 - Software is pervasive throughout the research lifecycle



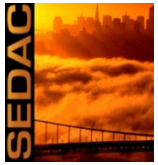
Software Used by Various Science Stakeholders



- Software is used for data collection, processing, and analysis.
- Methods, algorithms, and workflows are encoded into software programs and scripts.
- Software is used for storing, preparing, and distributing data to answer new questions and test new hypotheses.



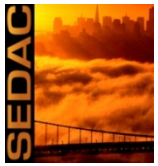
Can We Rely on Software in the Future?



- Software is an integral aspect of today's science infrastructure that will be relied upon in the future.
- If needed software is not available or cannot be reused, the future of science will suffer.
- Making science software sustainable is good science practice.



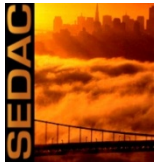
Which software artifacts should be sustainable?



- Source Code
- Compiled Code
- Scripts
- Code Generators
- Algorithms
- Plug-ins
- Servers
- Operating Systems
- Compilers
- Designs
- Architectures
- Frameworks
- System Diagrams
- Input Values
- Output Values
- Documentation
- User Guides
- Licenses



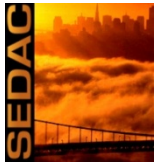
Software is Important for Science



- Software is important – it is used, reused, and being recognized as a contribution to science



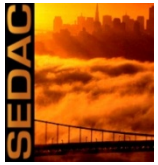
Software Contributions Need Recognition



- Similar to publications, data and software contribute to science
- Data and software contributions are being formally recognized in science
- If software is considered valuable, it may be managed



Software is Recognized in Grant Proposals

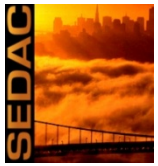


- NSF GPG Instructions for Results from Prior NSF Support
 - "evidence of research products and their availability, including, but not limited to: data, publications, samples, physical collections, software, and models". NSF 14-1, II-10.
- NSF GPG Instructions for Biographical Sketches - Products
 - "Acceptable products must be citable and accessible including but not limited to publications, data sets, software, patents, and copyrights". NSF 14-1, II-12.
- NSF GPG Instructions for Data Management Plans
 - "types of data, samples, physical collections, software, curriculum materials, and other materials to be produced in the course of the project" NSF 14-1, II-21.

Source: The National Science Foundation Proposal and Award Policies and Procedures Guide: Part 1 - Grant Proposal Guide. 2013. Effective February 24, 2014. NSF 14-1.



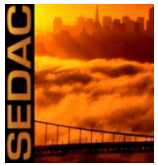
Opportunities to Recognize Software Contributions



- Share software with others
- Alternative Metrics (Altmetrics)
 - Software contributions and their uses count
- NIH biosketch emphasizes accomplishments instead of just publications
- See <http://nexus.od.nih.gov/all/2014/05/22/changes-to-the-biosketch/>



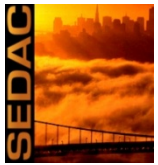
Some Impediments to Sustainable Software



- Maintenance costs and efforts
- Obsolescence
- New versions
- Incompatible platforms
- Restrictive Licenses
- No reuse opportunities



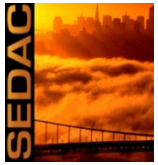
Key Practices for Creating Sustainable Software



- Documentation
- Open licenses
- Software packaging
- Packaging software with data
- Archiving
- Citing software



Reuse Readiness Levels (RRLs)



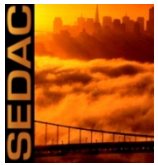
- RRLs specify metrics for reusability of software
- Nine metrics are defined within nine levels
- Reusability of software contributes to sustainability

Source: NASA Earth Science Data Systems Software Reuse Working Group (2010). Reuse Readiness Levels (RRLs), Version 1.0. April 30, 2010.

<https://earthdata.nasa.gov/esdswg/software-reuse/reuse-readiness-levels-rrls>



Nine RRL Topic Areas

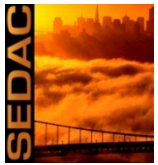


- Documentation
- Extensibility
- Intellectual Property Issues
- Modularity
- Packaging
- Portability
- Standards compliance
- Support
- Verification and Testing

Source: NASA Earth Science Data Systems Software Reuse Working Group (2010). Reuse Readiness Levels (RRLs), Version 1.0. April 30, 2010.
<https://earthdata.nasa.gov/esdswg/software-reuse/reuse-readiness-levels-rrls>



Summaries of the RRLs

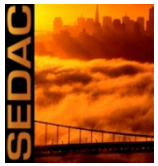


Level	Summary
RRL 1	Limited reusability; the software is not recommended for reuse.
RRL 2	Initial reusability; software reuse is not practical.
RRL 3	Basic reusability; the software might be reusable by skilled users at substantial effort, cost, and risk.
RRL 4	Reuse is possible; the software might be 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; the 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.

Source: NASA Earth Science Data Systems Software Reuse Working Group (2010). Reuse Readiness Levels (RRLs), Version 1.0. April 30, 2010.
<https://earthdata.nasa.gov/esdswg/software-reuse/reuse-readiness-levels-rrls>



RRL Example: Verification and Testing



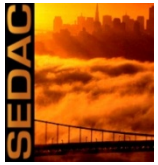
RRL 5 – Software application tested and validated in a laboratory context.

The fidelity of the software application testing has not been demonstrated. The software application must be integrated with reasonably realistic supporting elements so that the total application (component level, sub-system level, or system level) can be tested in a “simulated” or somewhat relevant context. At this level, issues such as scalability, load testing, and security are addressed when applicable.

Source: NASA Earth Science Data Systems Software Reuse Working Group (2010). Reuse Readiness Levels (RRLs), Version 1.0. April 30, 2010.
<https://earthdata.nasa.gov/esdswg/software-reuse/reuse-readiness-levels-rrls>



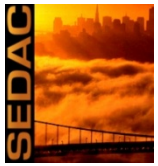
Planning for Software Sustainability



- Decide early on approaches to be used
- Determine whether software may be Open Source
- Identify archive for preserving software
- Establish target goals for achieving sustainability



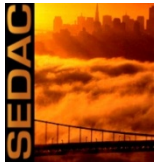
Include Software Sustainability in Science Education



- Formal science instruction
 - K-16
 - Graduate programs in science
- Professional and workforce development
 - Workshops and training opportunities
 - Webinars and Demonstrations
 - On Demand Self-Paced Learning Opportunities



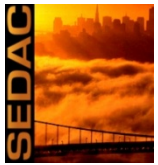
Software is Science Infrastructure



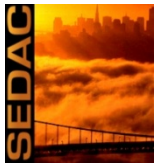
- If the artifacts of the study are in digital form, then the research is dependent on software!



Software Maintenance



- When maintaining facilities and equipment, consider software maintenance as well.



-
-
-
- When developing software, think about the user. It could be you, tomorrow!