



# Software Reuse to Support Earth Science

*James J. Marshall (Innovim / NASA GSFC)*

Robert R. Downs (CIESIN, Columbia University)

Shahin Samadi (Innovim / NASA GSFC)

Neil S. Gerard (Innovim / NASA GSFC)

Robert E. Wolfe (NASA GSFC)

DSADR 2008 (Domain Specific Analysis and Design for Reuse)

Workshop held in conjunction with ICSR 2008

May 25, 2008



NATIONAL AERONAUTICS  
AND SPACE ADMINISTRATION

## **Background Information**

About the Reuse Working Group

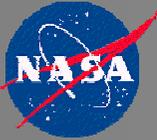


- About the NASA Earth Science Data Systems (ESDS) Software Reuse Working Group (WG):
  - The WG was started in 2004 to facilitate reuse of software assets within the NASA Earth science community.
  - Membership is limited to NASA-funded projects and investigators, though there have been many contributions from the general Earth science community.
  - The WG has been working to establish a “marketplace” for reusable Earth science software artifacts by working to increase the supply and availability of reusable assets.
  - Also, the WG has worked to increase the community capacity and desire for reuse by demonstrating the feasibility and value of reuse.
  - Through regular meetings of the full WG and a smaller support team, a variety of activities are performed to encourage and enable reuse.
- Goals of the Reuse WG include:
  - To spend less time, money, and effort on software development
  - To increase productivity and improve quality through reuse
  - To increase the number of available reusable assets



## Reuse WG Charter Highlights

- Purpose
  - Address technical issues required to enable and facilitate reuse of software assets, including open source products, within the NASA Earth science community
- Goals
  - Demonstrate the feasibility and value of reuse
  - Increase the supply and availability of reusable assets
  - Make recognizable and easy-to-evaluate candidate reuse solutions
  - Minimize the cost of infrastructure activities to support the community's reuse activities
  - Increase community capacity and interest in reusing existing assets
  - Contribute to the removal of existing barriers to reuse
  - Recommend incentives to encourage reuse
- Scope
  - Facilitating reuse across projects and not interfering with local control of participating systems
  - Focusing on reuse process and not on technology infusion process
  - Focusing on reuse of existing assets rather than reusability of newly developed assets
  - Focusing not only on software code, but also on design artifacts (architectures, software designs, ICDs, test plans, etc.)
  - Focusing on reuse of proven operational and NASA Earth science specific software assets



## Reuse WG Activities

### Reuse Implementation Projects

Efforts that result in the publication or use of a reusable component

### Support/Enablement Activities

Efforts that provide tools and mechanisms to enable reuse

### Outreach and Education Activities

Efforts that increase community awareness and understanding of benefits, best practices, etc.

### Policy Change Activities

Efforts to reduce policy barriers to reuse

### Reuse Incentive Activities

Awards and structural changes that directly or indirectly encourage reuse

- Examples of work in some of these areas include:

- Recommending that NASA create a **Reuse Enablement System** (repository) for Earth science reusable software assets; development of **Reuse Readiness Levels**
- Creating a **web site to promote and provide information about reuse**
- Providing NASA with **policy recommendations to encourage reuse**
- Developing a **reuse peer-recognition award**



## **Software Reuse Portal Web Site**

An education and outreach activity



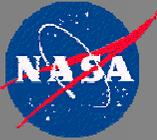
- Key drivers for the web portal include:
  - Serve the community of Earth science data systems and software developers who are interested in reuse
  - Serve as a gateway for reuse information relevant to the community
  - Establish a portal for the community to share resources on reuse
  - Distribute various resources on reuse to the community
  - Foster easier access to resources on reuse
- Major content categories based on purposes identified for the web portal include:
  - List of catalogs of reusable assets, tools, etc.
  - Reference library including events, news, Working Group documents, guidelines, and other resources
  - Information on open source software projects and licensing
  - Funding opportunities within and outside NASA
- “Suggest content” feature for user-submitted ideas



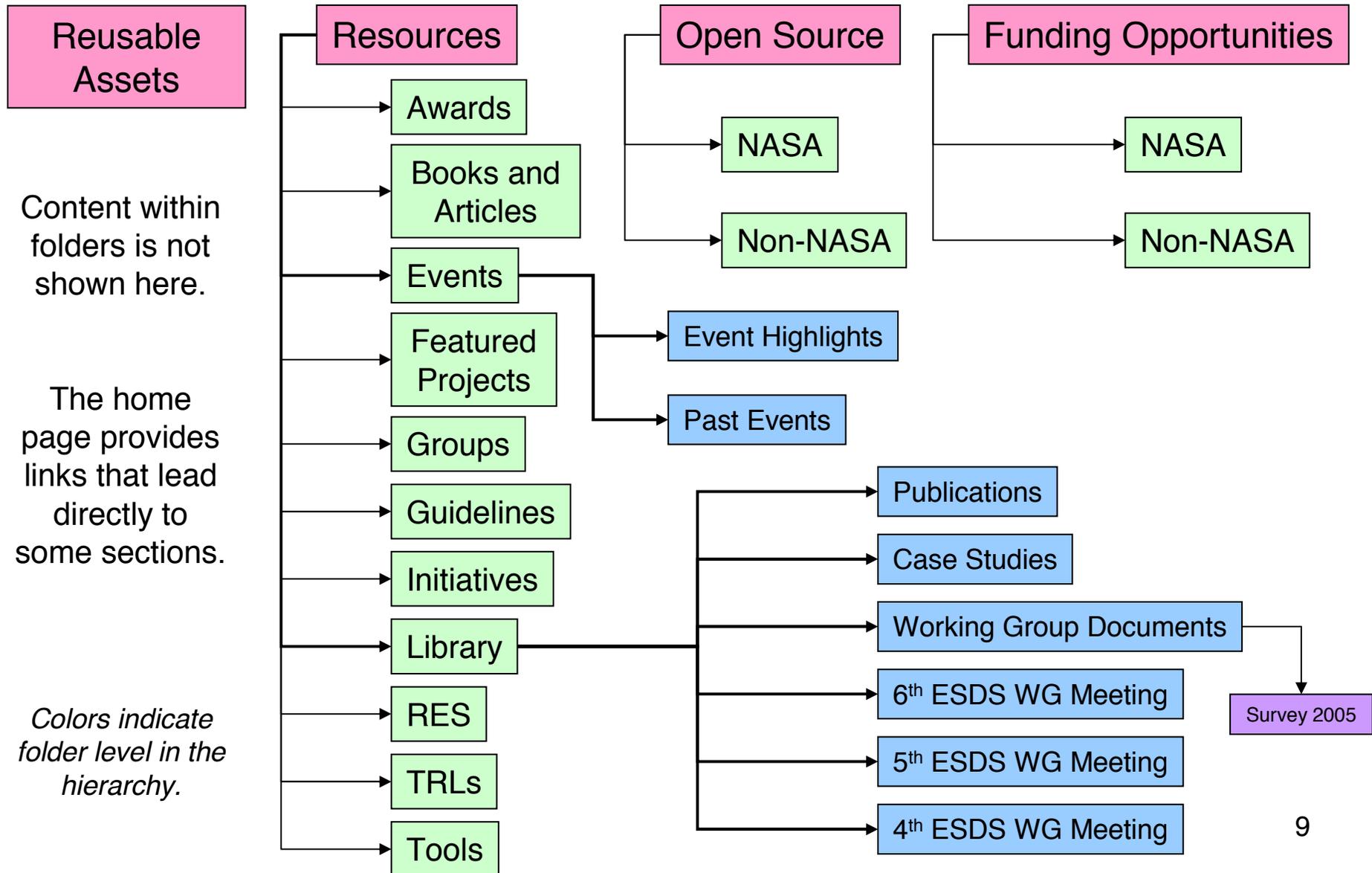
## Details on the Portal

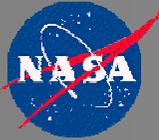
- Basic development
  - Built using the open source products Plone and Zope
  - Content organized into 4 main areas
- Basic web stats, 12/05 to 4/08:
  - Over 18000 visits by more than 13000 unique visitors, including almost 1700 repeat visitors
  - Nearly 61000 page views
  - Average ~630 visitors per month
  - Site has been in top 3 hits for “software reuse” on major search engines, and still achieves high placement in search results
  - Had Google PageRank 6 before site changed its domain name.

The screenshot shows the Earth Science Data System Software Reuse Working Group portal. The navigation menu at the top includes links for '+ HOME', '+ REUSABLE ASSETS', '+ RESOURCES', '+ OPEN SOURCE', and '+ FUNDING OPPORTUNITIES'. The main content area features a 'news' section with several articles, an 'upcoming events' section, and a 'Demand' section with a graphic of people walking on a path. A red arrow points from the 'REUSABLE ASSETS' link in the navigation menu to the 'news' section.



# Portal Folder Hierarchy



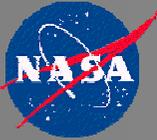


- Resources
  - Books and articles: suggested by WG members or portal visitors, and listed after reviews by WG members agree that they are relevant
  - Events: listing of meetings and conferences on reuse and Earth science, suggested by WG members or portal visitors, and listed after reviews by WG members agree that they are relevant
  - Library of WG Material: includes presentations, publications, case studies, and other items created by WG members
  - Guidelines: a number of short articles written by WG members on topics in bottom-up reuse and technology transfer
- Reusable Assets
  - Links to software asset collections of interest to the Earth science community (e.g., GCMD, ECHO, HDF tools)
  - Links to NASA open source software and information

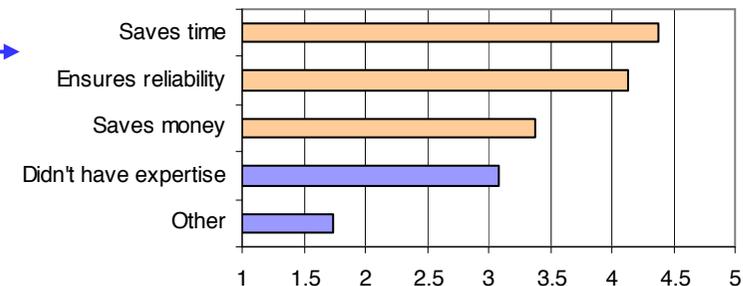


## **Reuse Enablement System (RES)**

A support and enablement activity



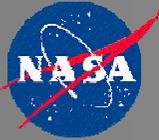
- A survey on the reuse practices of the Earth science community was conducted in 2004 and repeated in 2005 with OMB approval and a wider audience. (Marshall et al., 2006)
- Both surveys show the same basic results:
  - Developers need to be able to easily locate and evaluate available reusable artifacts.
  - Top three motivations for reuse → match the WG goals:
    - Saving time
    - Ensuring reliability
    - Saving money
  - Top three factors to increase reuse:
    - Earth science catalog/repository of reusable assets
    - Greater use of open source licensing
    - More education and guidance on reuse
  - Top two barriers to reuse:
    - Did not know reusable assets existed
    - Did not know where to look for reusable assets



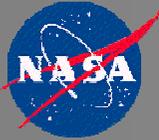
Areas where  
the WG can  
provide help.



- Based on the first survey results, the WG recommended that an Earth science repository/catalog system be created to meet the needs of the Earth science software developer community.
  - Having a catalog/repository for reusable Earth science assets is one of the best means of increasing reuse in the community.
  - It would also address the top two barriers to reuse.
- During the same period of time, the WG developed use cases and requirements for the proposed Reuse Enablement System.
- In response to the recommendation, NASA Headquarters tasked the WG to perform a trade study to understand the role of existing systems as a potential platform for enabling software reuse.



- The WG conducted a trade study of various NASA and non-NASA sites.
- The results showed that none of the existing systems satisfied the needs of the community of Earth science software developers.
- The WG then conducted an architecture study to determine what existing software package/system was most suited for reuse in building the RES.
- The results showed that the XOOPS content management system met the most requirements and would take the least time to develop.
- The WG began work on developing a prototype RES built off the XOOPS package, adding to it and modifying it as necessary to meet all of the RES requirements.
- Currently, the WG is developing a test plan for formal testing of the prototype, and plans to provide the prototype RES to the NASA community for their use.
- The WG is also working on a set of policies for the operation and maintenance of the RES.



- As part of the recommendation, the WG estimated the size of the target audience for the RES and the number of assets the RES might contain.
- Target audience:
  - NASA's workforce consists of about 82 000 people.
  - About 4% of them are in Earth science and of those, about 60% are scientists and engineers.
  - Therefore, a lower limit to the audience is about 2 000 people.
- Number of assets:
  - SourceForge contains about 172 000 projects and 1 800 000 users.
  - Assuming one unique provider per project, ~10% of users are providers, and this estimate is used for the RES.
  - Based on the above audience estimate, if 10% are providers and each offers 1 asset, the RES could contain at least 200 assets.

*The peer-recognition award being developed by the WG will provide an incentive for users to contribute assets to the RES.*



# Views of the Prototype (1 of 3)

**Home page as viewed by an Anonymous User**

*Consumers would also see this note on the main downloads page:*

**Main Downloads page as viewed by a Provider**

*Categories are not final*

**Note:** You currently do not have the ability to upload files. If you would like to do so, find out how to [become a provider!](#)



# Views of the Prototype (2 of 3)

The screenshot shows the NASA ESDS Reuse Working Group website. The top navigation bar includes links for 'View Account', 'Edit Account', 'Notifications', 'Inbox', and 'Logout'. Below this is a search bar and a 'Search NASA' button. The main content area features a 'Downloads' section with a navigation menu on the left. The 'Downloads' page is for a category, showing a search bar, a 'Browse Downloads by alphabetical listing' link, and a grid of download links. A 'Category Description' section indicates 'Java functions'. Below this is a red-bordered box containing 'Active Notifications' and 'Notification Options' for the category, with checkboxes for 'Global' and 'Category' notifications and an 'UPDATE NOW' button. The footer contains a privacy policy link, the NASA logo, and curator information.

Downloads page for a category (Provider view)

The screenshot shows the full detail page for an asset named 'calc'. The page includes a breadcrumb trail: '+ Home > Downloads > Algorithms > Perl'. The asset title is 'calc'. The description states: 'This web page offers a calculator Perl script that includes mathematical functions and pre-defined constants commonly used in astronomy.' The submitter is 'RES Admin', released on 'Mon, 15-Oct-2007'. Other details include 'Version: 0', 'Downloads: 12', 'File Size: 8.00 KB', 'Platform: None', and 'Checksum: None'. The home page is 'Astronomy aware Unix Calculator'. A rating section shows '2 Votes'. Download times are listed: 'Modem(56k) : 1s', 'ISDN(64k) : 1s', 'DSL(768k) : 0.09s', and 'LAN(10M) : 0.01s'. A red-bordered box highlights the 'Rate Resource | Report Broken | Modify | Recommend | Comments (1)' links. Below this is a list of 'Other files by: RES Admin' including 'test download', 'Perl Limericks', 'coord\_conv.pl', 'circle.c', 'calc', 'linept.c', 'linreg.c', and 'ESDS Software Reuse Working Group Year End Report 2006'. A 'Comments' section shows a dropdown menu set to 'Nested' and 'Oldest First', with '+ REFRESH' and '+ POST COMMENT' buttons. The comment text reads: 'The comments are owned by the poster. We aren't responsible for their content.' A table at the bottom shows the poster 'marshall' joined on '2008/3/21' and the thread 'Re: calc' with a comment: 'I've used this tool since grad school. It's useful as a general'.

Full detail page for an asset (Provider view)



# Views of the Prototype (3 of 3)

Control Panel Home | XOOPS News

Logout | Home Page

Downloads Admin

Preferences | Permissions | Block Settings | Go to module | About

Main Index | Index Page Management | Category Management | Create new Download | Upload Picture | Mimetypes Management

Vote Data | Statistic | Upgrade and Import Tool

**Module Admin Summary**

Category: 13 | Files: 14 | Submitted: 0 | Modified: 0 | Broken: 1 |

**Server Status**

**Information taken from PHP ini File:**

- GD Library Support: **Enabled** (Thumbs Nails Available)
- GD Library Version: **2.0 or higher**
- Safe Mode Status: **OFF**
- Register Globals: **ON**
- Server Uploads Status: **ON**
- Max Upload Size Permitted: **2M**
- Max Post Size Permitted: **8M**
- Max Time for running scripts: **30 s**

**System Configuration**

Avatars	Banners	Blocks	Comments	Find Users
Groups	Image Manager	Mail Users	Modules	Preferences
Smilies	Templates	User Ranks	Edit Users	

**Information about the upload-paths**

Uploads will be stored in this folder: /var/www/reuse/alpha/res/htdocs/uploads/PDdownloads/ ( available )  
 Category Images will be stored in this folder: /var/www/reuse/alpha/res/htdocs/modules/PDdownloads/images/ ( available )  
 Screenshot Images will be stored in this folder: /var/www/reuse/alpha/res/htdocs/modules/PDdownloads/images/screenshots/ ( available )

**Modify Category**

Category Title:

Modify Delete

**Published Files:**

ID	File Title	Poster	Submission Date	Published	Action
17	Life Cycle Statement	marshall	2008/4/8	●	
15	test download	RES Admin	2007/10/24	●	
13	Perl Limericks	RES Admin	2007/10/16	●	
12	coord_conv.pl	RES Admin	2007/10/16	●	
11	circle.c	RES Admin	2007/10/16	●	
10	Average Inputs	Neil_Provider	2007/10/16	●	

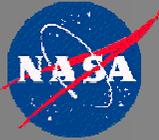
Administrator main menu for basic system configuration

Administrator menu for Downloads module configuration



## **Reuse Readiness Levels (RRLs)**

A support and enablement activity that also serves as education and outreach

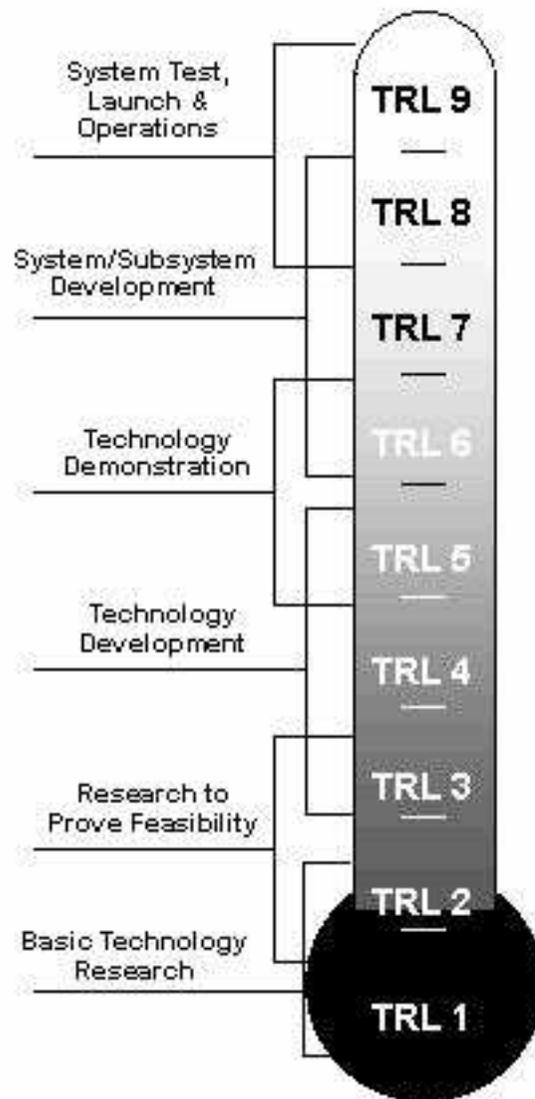


- Technology Readiness Levels (TRLs) and similar measures can be used to evaluate the maturity of a particular technology.
  - NASA TRLs range from 1 to 9, going from basic principles to mission proven.
  - While designed more for hardware, these have also been applied to software (see next slide).
- These measures typically do not consider reuse/reusability, or do so only in a limited manner.
  - The emphasis is the maturity of the technology as a whole.
  - The Open Process Framework's Technology Readiness Assessment is one of the few that includes reuse, but only in terms of reused critical technologies.

# Technology Readiness Levels

## Applied to Software

(v4 5/6/99)



### TRL 9: Actual system “flight proven” through successful mission operations

*Thoroughly debugged software. Fully integrated with operational hardware/software systems. All documentation completed. Successful operational experience. Sustaining software engineering support in place. Actual system fully demonstrated.*

### TRL 8: Actual system completed and “flight qualified” through test and demonstration (Ground or Flight)

*Thoroughly debugged software. Fully integrated with operational hardware and software systems. Most user documentation, training documentation, and maintenance documentation completed. All functionality tested in simulated and operational scenarios. V&V completed.*

### TRL 7: System prototype demonstration in a space environment

*Most functionality available for demonstration and test. Well integrated with operational hardware/software systems. Most software bugs removed. Limited documentation available.*

### TRL 6: System/subsystem model or prototype demonstration in a relevant environment (Ground or Space)

*Prototype implementations on full scale realistic problems. Partially integrated with existing hardware/software systems. Limited documentation available. Engineering feasibility fully demonstrated.*

### TRL 5: Component and/or breadboard validation in relevant environment

*Prototype implementations. Experiments with realistic problems. Simulated interfaces to existing systems.*

### TRL 4: Component and/or breadboard validation in laboratory environment

*Standalone prototype implementations. Experiments with full scale problems or data sets.*

### TRL 3: Analytical and experimental critical function and/or characteristic proof-of-concept

*Limited functionality implementations. Experiments with small representative data sets. Scientific feasibility fully demonstrated.*

### TRL 2: Technology concept and/or application formulated

*Basic principles coded. Experiments with synthetic data. Mostly applied research.*

### TRL 1: Basic principles observed and reported

*Basic properties of algorithms, representations & concepts. Mathematical formulations. Mix of basic and applied research.*

Source: <http://isd.gsfc.nasa.gov/Technology/TRL/TRL.ppt>

Comments to [kswanson@mail.arc.nasa.gov](mailto:kswanson@mail.arc.nasa.gov)

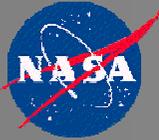


- The issue of how to measure the maturity of software—in a reusability sense—was discussed at the 2006 ESDSWG Meeting.
- Having a measure of the reusability of an asset:
  - Provides potential reusers with additional information about the reuse maturity of the asset.
    - It lets them know what they're getting, and
    - Gives them a basic feel for what modifications may be needed.
  - Helps potential reusers make better informed choices about
    - What to reuse, and
    - What best meets their needs.
- *This measure can be used as a piece of metadata for assets placed in a Reuse Enablement System (RES) or anywhere else.*
- The Software Reuse WG is developing a set of Reuse Readiness Levels (RRLs) to measure the maturity of a technology with respect to reusability.



- Through discussions on weekly and monthly telecons, the Software Reuse WG made the following decisions:
  - To use nine levels, to align with the familiar TRL scale.
  - To look at nine topic areas that the WG felt were important for measuring the reuse maturity of software.
- Volunteers from the WG:
  - Wrote an initial set of levels for each topic (2+ people per topic),
  - Drafted summaries of each RRL, looking at the levels for all topic areas,
  - Created a set of summary RRLs by combining information from all topics at the same level, and
  - Made suggested revisions based on feedback received from the community.

*Note: the RRLs presented here are still under development.*



### Topic areas included:

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

### Example from Verification and Testing

**RRL 4** – Software application tested and validated in laboratory environment

Following successful testing of inputs and outputs, the testing would include integrating an application to establish that the “pieces” will work together to achieve concept-enabling levels. This validation must be devised to support the concept that was formulated earlier and should also be consistent with the requirements of potential system applications. The validation is relatively “low-fidelity” compared to the eventual system: it could be composed of ad hoc discrete components in a laboratory; for example, an application tested with simulated inputs.



# Proposed RRL Topic Levels

Level	Documentation	Extensibility	Intellectual Property Issues	Modularity	Packaging	Portability	Standards Compliance	Support	Verification and Testing
1	Limited internal documentation available	No ability to extend or modify program behavior	Potential owners and stakeholders of product have been identified.	No designs for modularity or reuse	Source code available	The software is not portable at any cost	Follows no particular standard	No support available	No testing performed
2	Fully commented source code available	Prohibitive costs and efforts need to modify or extend the system	Relevant intellectual policies of potential owners and stakeholders have been reviewed.			Some parts of the software may be portable	Follows some parts of common standards and best practices	Known contact available	Software application formulated and unit testing performed
3	Basic external documentation available	Can be extended with the input of considerable time and effort on par with recreating system separately	Intellectual property agreements have been proposed to potential stakeholders.	Modularity at major system or subsystem level only	Detailed installation instructions available	The software is only portable with significant costs	Follows a company-wide standard for development and testing	Original developers provide proactive support	Testing includes testing for error conditions and proof of handling input errors
4	Reference manual available	Can be modified and extended through configuration changes, minimal modification of source	Potential stakeholders have negotiated on intellectual property agreements and authorship issues.			The software may be portable at a reasonable cost	Most components follow a complete, universal standard, but not validated	Latest updates or patches are available but not very frequently	Software application demonstrated in a laboratory environment
5	User manual available	Consideration for future extensibility designed into system, extensibility approach somewhat defined	Agreement and approval on authorship, attribution, and intellectual property issues has been obtained from stakeholders.	Partial segregation of generic and specific functionality	Software is easily configurable for different environments	The software is moderately portable	All components follow a universal standard, but only partially validated	Informal user community available	Software application tested and validated in a laboratory environment
6	Tutorials available	Designed from the start to allow easy extensibility, provides many points of extensibility and a thorough and detailed extensibility plan	Authorship, attribution, and intellectual property statements have been drafted to reflect agreement among stakeholders on intellectual property and authorship.			The software is portable	Validated to follow a specific proprietary standard	Centralized support available	Software application demonstrated in a relevant environment (Earth science related)
7	Interface guide available	Proven to be extensible internally, code structured to provide loose coupling and high cohesion	Authorship and intellectual property statements included in product prototype.	Clear delineations of specific and reusable components	OS detect and auto-build for supported platforms	The software is highly portable	Validated to comply to a specific open standard	Organized/defined support by the original developer available	Software application tested and validated in a relevant environment (Earth science related)
8	Extension guide and/or Design/Development guide available	Proven extensibility on a major external program, provides a clear plan for modifying and extending features	Manifestation of authorship, attribution, and intellectual property statements reviewed in product prototype before product release.				Proven by validation to comply with a "gold" standard	Support by organization available	Software application "qualified" through test and demonstration (meets requirements) and successfully delivered to the Earth science environment
9	Full software lifecycle engineering design documentation available	Proven extensibility in multiple scenarios, provides specific documentation and features to build extensions	Reviewed authorship, attribution, and intellectual property statements packaged with product for release.	All functions and data encapsulated into objects or accessible through web service interfaces	GUI installation environment provided	The software is completely portable	"Gold" standard compliance of entire system and development, independently validated	Large user community with well-defined support available	Actual software application tested and validated through successful use of application output



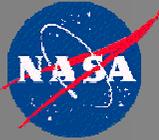
## Example 1: Documentation Topic Levels

Level	Documentation Level Summary
1	Limited internal documentation available
2	Fully commented source code available
3	Basic external documentation available
4	Reference manual available
5	User manual available
6	Tutorials available
7	Interface guide available
8	Extension guide and/or Design/Development guide available
9	Full software lifecycle engineering design documentation available



## Example 2: Intellectual Property Levels

Level	Intellectual Property Level Summary
1	Potential owners and stakeholders of product have been identified.
2	Relevant intellectual policies of potential owners and stakeholders have been reviewed.
3	Intellectual property agreements have been proposed to potential stakeholders.
4	Potential stakeholders have negotiated on intellectual property agreements and authorship issues.
5	Agreement and approval on authorship, attribution, and intellectual property issues has been obtained from stakeholders.
6	Authorship, attribution, and intellectual property statements have been drafted to reflect agreement among stakeholders on intellectual property and authorship.
7	Authorship and intellectual property statements included in product prototype.
8	Manifestation of authorship, attribution, and intellectual property statements reviewed in product prototype before product release.
9	Reviewed authorship, attribution, and intellectual property statements packaged with product for release.



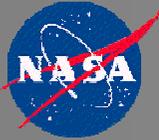
## Example 3: Support Levels

Level	Support Level Summary
1	No support available
2	Known contact available
3	Original developers provide proactive support
4	Latest updates or patches are available but not very frequently
5	Informal user community available
6	Centralized support available
7	Organized/defined support by the original developer available
8	Support by organization available
9	Large user community with well-defined support available



## Draft RRL Summaries

RRL	RRL Summary
1	No reusability – software is not reusable (or is not recommended for reuse)
2	Initial reusability – software reuse is not practical
3	Basic reusability – software might be reusable by skilled users at substantial effort, cost, and risk
4	Reuse is possible – software might be reused by most users with some effort, cost, and risk
5	Reuse is practical – software could be reused by most users with reasonable cost and risk
6	Software is reusable – software can be reused by most users although there may be some cost and risk
7	Software is highly reusable – software can be reused by most users with minimum cost and risk
8	Demonstrated reusability – software has been reused by multiple users
9	Proven reusability – software is being reused by many classes of users over a wide range of systems



# Draft RRL Descriptions

RRL	RRL Description
1	Little is provided beyond limited source code or pre-compiled, executable binaries. There is no support, contact information, author attribution, or rights specified, the software is not extensible, and there is inadequate or no documentation.
2	Some source code, documentation, and contact information are provided, but these are still very limited. Initial testing has been done, but authorship and reuse rights are still unclear. Reuse would be challenging and cost-prohibitive.
3	Software has some modularity and standards compliance, intellectual property agreements have been proposed, some support is provided by developers, and detailed installation instructions are available, but rights are unspecified. An expert may be able to reuse the software, but general users would not.
4	Software and documentation are complete and understandable. Software has been demonstrated in a lab on one or more specific platforms, infrequent patches are available, and intellectual property issues have been negotiated. Reuse is possible, but may be difficult.
5	Software is moderately portable, modular, extendable, and configurable, has low-fidelity standards compliance, a user manual, and has been tested in a lab. A user community exists, but may be a small community of experts. Authorship and rights are not specified.
6	Software has been designed for extensibility, modularity, and portability, but software and documentation may still have limited applicability. Tutorials are available, and the software has been demonstrated in a relevant environment. Intellectual property statements have been drafted, but authorship and rights have not been formalized.
7	Software is highly portable and modular, has high-fidelity standards compliance, provides auto-build installation, and has been tested in a relevant environment. Support is developer-organized, and an interface guide is available. Software and documentation are applicable for most systems.
8	Software has been shown to be extensible, and has been qualified through test and demonstration. An extension guide and organization-provided support are available. Intellectual property is reviewed in the product before release, and authorship and rights are specified.
9	Software is fully portable and modular, with all appropriate documentation and standards compliance, encapsulated packaging, a GUI installer, and a large support community that provides patches. Software has been tested and validated through successful use of application output. Complete and clear attribution and permissions for implementation by various user levels are available.

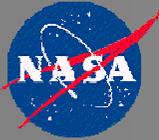


## Factors Under Consideration

- *Security*: Could this be incorporated into verification/testing, should it be its own topic area, or is it not a factor of reusability?
- *Use vs. reuse*: When is a factor more about how good it is for your application (use) than is it ready for you to use (reuse)? How much should use be considered? Higher reusability may sacrifice out-of-the-box usability.
- *Quantitative measures*: To make the ratings easier to determine, with less ambiguity, more objective level criteria are needed. Also, how to maintain consistency? Should (self-)assessments be audited and how?
- *Cost and Risk*: How to factor in these concerns? [Started to address them briefly in the RRL summary levels.](#)
- *Topic level ratings*: These are viewed as useful information for reusers, so how should the information be offered?
- *Target audience*: Should be clarified since engineers may prefer the nine topic levels, but project managers may prefer the overall summaries.
- *Various types of reuse*: How to consider things like black box reuse, white box reuse, reuse over long periods of time, reuse in virtual machines, etc.?
- *An RRL calculator*: To help providers and consumers rate the reusability of software assets. [A sample RRL calculator has been made for the RES.](#)



## Conclusion



- 
- NASA's ESDS Software Reuse WG is involved in a variety of activities to encourage and support the reuse of software in the Earth science community, including:
    - Software reuse portal web site
    - Reuse Enablement System (RES)
    - Reuse Readiness Levels (RRLs)
    - Policy change recommendations
    - Assisting others in reuse projects
    - Developing a peer-recognition award
  - Future activities include:
    - Continuing work on all of the above activities
    - Working to facilitate NASA's software release process and lower barriers for certain types of software, if possible
    - Continuing to promote software reuse through publications and presentations



- NASA Earth Science Data Systems (ESDS) Working Groups
  - Coordinator, Kathy Fontaine (<http://esdswg.gsfc.nasa.gov/>)
- Software Reuse Working Group
  - Chair, Robert E. Wolfe ([Robert.E.Wolfe@nasa.gov](mailto:Robert.E.Wolfe@nasa.gov))
  - Outreach and Education Team Leader: Robert R. Downs ([rdowns@ciesin.columbia.edu](mailto:rdowns@ciesin.columbia.edu))
  - General Information: James J. Marshall ([James.J.Marshall@nasa.gov](mailto:James.J.Marshall@nasa.gov))

The screenshot shows the website for the Earth Science Data System Software Reuse Working Group. The header includes the NASA logo and 'GODDARD SPACE FLIGHT CENTER'. Below the header is a navigation menu with links for HOME, REUSABLE ASSETS, RESOURCES, OPEN SOURCE, and FUNDING OPPORTUNITIES. The main content area includes a 'news' section with recent updates, a 'developing an internal marketplace for reusable software development artifacts' section, and a 'Demand' section with a graphic of people walking on a path.

**news**

- Reuse Enablement System (RES) Documents Added 2008-02-28
- 6th ESDS WG Meeting Reuse Materials Posted 2007-10-31
- Draft Agenda for 6th Joint ESDS Working Group Meeting Available 2007-09-19
- Prototype Internal NASA Reuse Enablement System in Development 2007-09-10
- Recent Publications Added 2007-08-20

**upcoming events**

- 10th International Conference on Software Reuse (ICSR10) Beijing, China, 2008-05-25
- AGU 2008 Joint Assembly Fort Lauderdale, Florida, 2008-05-27
- TeraGrid '08

**Earth Science Software Reuse**  
last modified 2008-02-28 08:43 AM

Welcome to the Earth Science Data System (ESDS) Software Reuse Portal. Please visit the [ESDS Working Group Page](#) for more information about our group.

We have posted some [reuse definitions](#) about what the Working Group does and does not consider reuse in our Resources, Library section, along with a [brochure](#) about our group and its activities. A number of [bottom-up reuse guidelines](#) are now available in the Resources, Guideline Documents section. We have also added a [technology transfer guideline](#) and [technology transfer FAQ](#) to help explain NASA's technology transfer process in the same section of the site. Some of our more recent [publications](#) have been added to the Resources, Library, Publications section, while the [S4PM case study](#) and the [SOSE case study](#) have been added to the Resources, Library, Case Studies section. A few new [non-NASA funding links](#) are on our Funding Opportunities page. Materials presented at the [6th ESDS WG Meeting](#) in October 2007 are now available, including some early work on drafting Reuse Readiness Levels (RRLs). The slides used for our discussion of RRLs at the [2008 Winter ESIP Meeting](#) are now available in the Resources, Events, Event Highlights section. Our full list of upcoming [events](#) is also in the Resources section.

We have completed our survey to better understand the practice of software reuse within the Earth Science community. Preliminary [survey findings](#) are available on the web and have been published in a [2006 IGARSS paper](#). We have also posted some of our documents related to our proposed [Reuse Enablement System \(RES\)](#) in a new folder under the Resources section. Our trade study of existing systems, use cases for the proposed RES, and requirements for the proposed RES are currently available.

For more information, please visit our [News](#), [Site Map](#), [About Us](#), [FAQ](#), [Suggest Content](#) and/or [Contact](#) page.

**Developing an internal marketplace for reusable software development artifacts**

**Demand**

Demonstrate the feasibility and value of reuse through focused projects  
Increase community capacity and desire (knowledge & tools) to reuse existing assets

It should be as easy to find a good quality reusable software asset as it is to find a book on the internet



## **Backup Slides**

Portal Web Site: 36–37

RES General: 38–42

RES Trade Study: 43–47

RES Requirements: 48–50

RES Architecture Study: 51–59

RRL Topic Levels: 60–66



# Views of the Portal (1 of 2)

NASA GODDARD SPACE FLIGHT CENTER

+ Visit NASA.gov  
+ Goddard Home Page

Earth Science Data System  
**Software Reuse**  
Working Group

+ HOME + REUSABLE ASSETS + RESOURCES + OPEN SOURCE + FUNDING OPPORTUNITIES

working group members [log in](#)

you are here: home → reusable assets

news

- New Awards Section  
2008-04-14
- Reuse Enablement System (RES) Documents Added  
2008-02-28
- 6th ESDS WG Meeting Reuse Materials Posted  
2007-10-31
- Draft Agenda for 6th Joint ESDS Working Group Meeting Available  
2007-09-19
- Prototype Internal NASA Reuse Enablement System in Development  
2007-09-10

More news...

upcoming events

- 10th International Conference on Software Reuse (ICSR10)  
Beijing, China, 2008-05-25
- AGU 2008 Joint Assembly  
Fort Lauderdale, Florida, 2008-05-27
- TeraGrid '08  
Las Vegas, Nevada,

Reusable Assets

Up one level

This section contains links to reusable software asset collections of interest to the Earth science community.

- Applied Information Systems Research Code and Algorithm Library** — last modified 2008-05-26 02:19 PM  
NASA's AISR Program created this site to archive and distribute software code, algorithms, packages, and related material with a space science focus.
- Computational Infrastructure for Geodynamics** — last modified 2005-05-26 03:08 PM  
CIG is working under a Cooperative Agreement with the National Science Foundation to develop, support, and disseminate open source software for the computational geophysics and related fields.
- Earth Observing System Clearinghouse** — last modified 2005-05-26 03:09 PM  
The ECHO Client Reuse page is a resource to enable the sharing of source code between ECHO client developers.
- Global Change Master Directory** — last modified 2005-07-19 04:46 PM  
NASA's GCMD enables users to locate and obtain access to Earth Science data sets and services relevant to the global change and Earth science research.
- HDF-EOS Tools and Information Center** — last modified 2006-11-09 05:24 PM  
A collection of libraries and tools for use with HDF-EOS.
- Sourceforge - Earth Science software** — last modified 2006-09-15 07:50 AM  
Earth Science software hosted on Sourceforge.net.
- Suggest a Link** — last modified 2007-05-30 10:23 AM  
Are there other catalogs or repositories that you think should be listed here?
- NASA Open Source** — last modified 2008-04-28 07:23 AM  
Several NASA centers also release software using the NASA Open Source Agreement.

Reusable Assets Section

you are here: home → resources

news

- New Awards Section  
2008-04-14
- Reuse Enablement System (RES) Documents Added  
2008-02-28
- 6th ESDS WG Meeting Reuse Materials Posted  
2007-10-31
- Draft Agenda for 6th Joint ESDS Working Group Meeting Available  
2007-09-19
- Prototype Internal NASA Reuse Enablement System in Development  
2007-09-10

More news...

upcoming events

- 10th International Conference on Software Reuse (ICSR10)  
Beijing, China, 2008-05-25
- AGU 2008 Joint Assembly  
Fort Lauderdale, Florida, 2008-05-27
- TeraGrid '08  
Las Vegas, Nevada, 2008-06-09
- 2008 NASA Earth Science Technology Conference (ESTC2008)  
College Park, MD, 2008-06-24
- 2008 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2008)  
Boston, Massachusetts, 2008-07-06

Resources

Up one level

- Suggest Content** — last modified 2007-05-04 02:41 PM  
Is there other content that you think should be listed here?
- Awards** — last modified 2008-04-14 10:25 AM  
Links to various awards for which reuse projects may be qualified.
- Books and Articles** — last modified 2007-05-31 08:18 AM  
Some suggested background reading. Books that are listed have been found relevant by one or more members of the working group. Articles that are listed have gone through a review process within the working group to assess their relevance to our site and audience. At least three people review each article, provide their reviews to an editor, and when a consensus is reached, the editor accepts/rejects the article for inclusion on the web site. Accepted articles are added to the site.
- Events** — last modified 2007-05-25 01:28 PM  
Upcoming conferences, conventions, and meetings
- Featured Projects and Technologies** — last modified 2007-05-09 08:12 AM  
Highlights of projects that employ software reuse and information about reuse in new and emerging technologies
- Groups** — last modified 2007-05-09 08:10 AM  
Other groups involved with software reuse.
- Guideline Documents** — last modified 2007-09-20 11:22 AM  
A set of guidelines on various topics in the area of bottom-up reuse, most based on presentations from the 5th ESDS WG Meeting, and other areas relevant to software reuse.
- Initiatives** — last modified 2005-07-29 01:47 PM  
Programs, standards, and other initiatives that support software reuse.
- Library** — last modified 2005-05-27 02:42 PM  
Downloadable working group publications, case studies, and other documents
- Reuse Enablement System (RES)** — last modified 2008-02-28 08:35 AM  
Documents related to the WG's proposed Reuse Enablement System (RES) for encouraging and enabling reuse in Earth science.
- Technology Readiness Levels** — last modified 2006-05-09 10:14 AM  
Links to information about Technology Readiness Levels (TRLs) and similar, alternative measures of technology maturity.
- Tools** — last modified 2005-05-27 02:11 PM  
Commonly used tools for enabling software reuse

Resources Section



# Views of the Portal (2 of 2)

NASA GODDARD SPACE FLIGHT CENTER + Visit NASA.gov + Goddard Home Page

Earth Science Data System  
**Software Reuse**  
Working Group

+ HOME + REUSABLE ASSETS + RESOURCES + OPEN SOURCE + FUNDING OPPORTUNITIES

working group members [log in](#)

you are here: home → open source

- news
- New Awards Section 2008-04-14
  - Reuse Enablement System (RES) Documents Added 2008-02-28
  - 6th ESDS WG Meeting Reuse Materials Posted 2007-10-31
  - Draft Agenda for 6th Joint ESDS Working Group Meeting Available 2007-09-19
  - Prototype Internal NASA Reuse Enablement System in Development 2007-09-10
- More news...

- upcoming events
- 10th International Conference on Software Reuse (ICSR10) Beijing, China, 2008-05-25
  - AGU 2008 Joint Assembly Fort Lauderdale, Florida, 2008-05-27
  - TeraGrid '08 Las Vegas, Nevada, 2008-05-28

**Open Source**  
last modified 2007-05-04 02:41 PM

**Open source or open source software (OSS) is any computer software distributed under a license which allows users to change and share the software freely. Open source software is required to have its source code freely available and end-users have the right to modify and redistribute the software to others.**

Open source projects support a collaborative development model whereby developers from multiple organizations can read, modify, and redistribute the source code. In the process of doing this the software evolves as people improve it, adapt it, and fix bugs. Proponents of the open source development model claim that, in many ways, it is superior to the closed source approach. An active open source development community can develop software with superior stability, reliability, and security to that developed using other development models. This collaborative development model can also be particularly attractive to the scientific community where scientific advances are often the result of the combined efforts of researchers in many different organizations.

The Reuse Working Group has identified traditional software licensing mechanisms as a potential barrier to software reuse. Traditional licensing typically requires the reuser to negotiate usage terms and conditions with the intellectual property owner every time that they want to reuse something. This can often be a time very consuming and arduous process. Greater use of open source licensing will help simplify the dissemination of software to the scientific community and promote a more collaborative development model for scientific software.

We recognize that open source licensing is not appropriate for all software. For example, the presence of proprietary code or export control restrictions may be valid reasons for choosing not to go open source. However, where proprietary technology or other intellectual property considerations preclude open source licensing, we would encourage you to make software available for reuse by the community using some other licensing mechanism that does not necessarily expose the source code.

For more information:

- NASA Open Source Links
- Non-NASA Open Source Links
- NASA Open Source Agreement

## Open Source Section

NASA GODDARD SPACE FLIGHT CENTER + Visit NASA.gov + Goddard Home Page

Earth Science Data System  
**Software Reuse**  
Working Group

+ HOME + REUSABLE ASSETS + RESOURCES + OPEN SOURCE + FUNDING OPPORTUNITIES

working group members [log in](#)

you are here: home → funding opportunities

- news
- New Awards Section 2008-04-14
  - Reuse Enablement System (RES) Documents Added 2008-02-28
  - 6th ESDS WG Meeting Reuse Materials Posted 2007-10-31
  - Draft Agenda for 6th Joint ESDS Working Group Meeting Available 2007-09-19
  - Prototype Internal NASA Reuse Enablement System in Development 2007-09-10
- More news...

- upcoming events
- 10th International Conference on Software Reuse (ICSR10) Beijing, China, 2008-05-25
  - AGU 2008 Joint Assembly Fort Lauderdale, Florida, 2008-05-27
  - TeraGrid '08 Las Vegas, Nevada, 2008-05-28

**Funding Opportunities**  
last modified 2008-01-24 05:22 PM

Both *Earth Science* and *Software Reuse* are areas of increasing research. Funding large-scale projects can require funding from multiple sources. Compiled are two directories dedicated to sources of research funding.

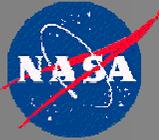
Funding Opportunities are divided into:

- NASA: [Funding opportunities within NASA](#)
- Other: [Funding opportunities outside NASA](#) including government or private sector grants.

Please let us know if there are any other funding opportunities that should be listed here:

- [Suggest a Link](#)

## Funding Opportunities Section



- **Focusing on Science**
  - The community is asking for a Reuse Enablement System (RES) to develop their applications faster, cheaper, and more reliably.
- **Supporting Education and Public Outreach**
  - The RES can make NASA software available to the public for education and other applications faster and more efficiently.
- **Complying with Standards**
  - Availability of a Reuse Enablement System is a core requirement for the Capability Maturity Model Integration (CMMI) and Federal Enterprise Architecture (FEA) compliance initiatives that NASA is working on.
- **Advancing Technologies**
  - The RES can enable new technologies such as Service Oriented Architecture (SOA) by making domain-specific services available to the community.



- From feedback received and discussions within the WG support team, some policy areas may include:
  - Intellectual property issues for uploads: how do we make sure the RES has permission to distribute uploaded assets?
  - Intellectual property issues for downloads: how do we make sure users of the RES can use and possibly modify or sell downloaded assets?
  - Upload permissions: who is allowed to upload assets?
  - Appropriateness of uploads: how to ensure that uploaded assets are valid and useful to the community?
  - Modification permissions: how should modifications to assets that are not one's own be handled?
  - Handling abuse of the system: what constitutes abuse and what actions should be taken to prevent or repair abuse?
  - Deprecation of assets: under what conditions is it appropriate to deprecate assets?
  - Version control: how will the RES handle multiple versions of the same asset?



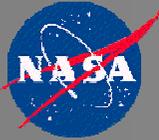
- 
- **Creating and Modifying Downloads**
    - Administrators can modify any user's download through administrator-only pages.
    - Providers can modify their own assets, with the changes taking affect after administration has approved the changes through administrator-only pages.
    - Provider can modify another user's assets with:
      - That user's approval
      - Administrator approval after original user's approval
    - If the user is only modifying features of a download, the RES will only update the original download. It should not create a new download or change the submitter name.



- **Role of Content Managers**
  - The Content Managers (site administration) are a specific group of users within the RES responsible for the daily operation. They are responsible for registering new users with the system, granting or denying provider access to users requesting it, approving or denying new assets, resolving reports of broken assets, deprecating or removing assets, filtering asset comments, and answering/relaying comments about the RES to the appropriate parties.
  - Each category within the downloads section of the RES may have a dedicated Content Manager(s). These users will be responsible for regulating the assets within the category assigned to them, and providing approval to new assets within that category.



- Privacy Information
  - The RES uses session cookies to store all session information. These are only used for login purposes.
  - IP addresses are recorded for all comments to downloads and reporting of broken links to track inappropriate use of the system.
  - We do not divulge any personal information to third parties.
  - We make every attempt to secure user passwords. All user passwords are stored as an encrypted 32 character hexadecimal string in the database.
  - The RES conforms with all NASA privacy policies and accessibility standards. For more information about these policies, please visit [http://www.nasa.gov/about/highlights/HP\\_Privacy.html](http://www.nasa.gov/about/highlights/HP_Privacy.html)



- 
- NASA sites reviewed:
    - Global Change Master Directory (GCMD)
    - Goddard Space Flight Center (GSFC) Open Source Software page
    - Ames Research Center Open Source Software page
    - HDF-EOS Tools and Information Center
    - Computational Technologies (CT) Project
    - Earth Observing System Clearinghouse (ECHO)
    - Planetary Data Systems (PDS) Software Download

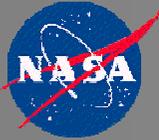


- Non-NASA sites reviewed:
  - Open Channel Foundation (hosts NASA's COSMIC Collection)
  - SourceForge
  - Freshmeat
  - Scientific Applications on Linux
  - National Technology Transfer Center
  - National HPC Software Exchange
  - Netlib
  - Savannah
  - Space Telescope Science Institute (STScI) Software and Hardware Products
  - Astronomical Software and Documentation Service



- NASA sites:
  - Direct Readout Laboratory
  - Glenn Research Center Software Repository
- Non-NASA sites:
  - ArcScripts
  - Wikipedia
  - Usenet newsgroups
  - Ruby Application Archive
  - SciRuby
  - Comprehensive Perl Archive Network (CPAN)
  - FreeGIS

In general, these sites were too narrowly focused to warrant a detailed review.



- 
- None of the existing operational sites fulfill the role of a software repository for the Earth science community.
  - None of the systems that were evaluated (as-is) sufficiently meet the requirements that are necessary to serve the community of Earth science software developers.
  - Shortcomings of existing systems include the following:
    - Not meeting enough of the critical functional requirements
    - Not focusing on the Earth science domain
    - Not targeting software developers as the primary audience
    - Not providing the type of small-sized assets that are most desired by the community of Earth science software developers for reuse purposes



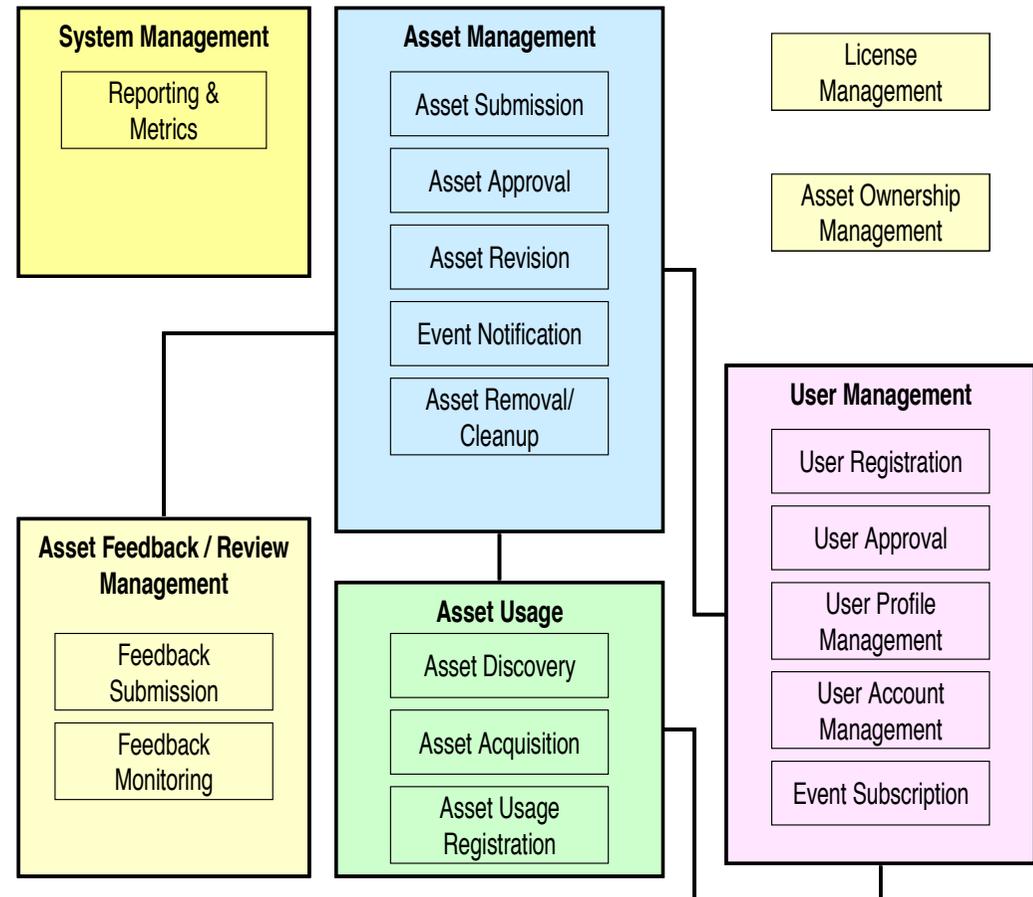
## Trade Study Conclusions

- A domain-specific catalog/repository system is needed to encourage and better enable software reuse within the community of Earth science software developers.
  - The GCMD is primarily a catalog of metadata, providing access to data sets, and the system is not designed to be used as a software repository.
  - Similarly, ECHO is middleware acting as a data/service broker.
  - The NASA Open Source Agreement sites have no real catalog or repository functionality and are restricted to NASA open source products.
  - Non-NASA sites are typically not domain-specific enough to meet the needs of a focused community.
- Some collaboration with existing systems may be possible, but existing systems alone cannot meet the needs of this community.
- Existing tools like the SourceForge software can be used in developing a reuse enablement system.
- Existing domain-specific reusable artifacts in other catalogs and repositories can be linked to by the RES.

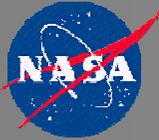


# RES Requirements Overview

- The Reuse Working Group collaborated for several months in 2004 to identify the important functional requirements needed for a Reuse Enablement System (RES), as illustrated in the figure.
- Additional functional requirements:
  - Minimal Operation Support
  - Performance
  - Security
  - Technology
- Important non-functional requirements:
  - Domain (Earth science focus)
  - Type of assets provided (small-sized components)
- Primary users of a RES are NASA-funded software developers within the Earth science community who create software products.



*See the Reuse Enablement System (RES) Requirements document revised May 7, 2007, for detailed descriptions of the requirements.*



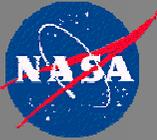
## Formalized Requirements

- The requirements developed by the Working Group and used in the Trade Study were revisited in summer 2006 to formalize them for use in the Architecture Study.
- Requirements were categorized into four major areas:
  - Users and User Information
  - Asset Storage and Management
  - Send and Manage Notifications
  - System Operations
- Each major category has a number of sub-categories for further classification of the requirements.
- In spring 2007, the requirements were re-titled and regrouped for clarity.
- The result was 54 requirements that are summarized in the RES Architecture Study and detailed in the RES Requirements document.
- Our use cases were similarly formalized in summer 2006.



# Reuse Enablement System Requirements

1. Users and User Information
  - 1.1. Support of User Types
    - 1.1.1. Consumer
    - 1.1.2. Provider
    - 1.1.3. Administrator
    - 1.1.4. Content Manager
  - 1.2. User Information Storage
    - 1.2.1. Common User Information
    - 1.2.2. Provider Information
  - 1.3. User Interface
    - 1.3.1. Profile Management
    - 1.3.2. Request Account Deletion
2. Asset Storage and Management
  - 2.1. Asset Information Storage
    - 2.1.1. Asset Information
    - 2.1.2. Asset Resources
    - 2.1.3. Asset Versions
    - 2.1.4. Asset Uploads
  - 2.2. Asset Discovery
    - 2.2.1. Alphabetic Listing
    - 2.2.2. Search
    - 2.2.3. Hierarchical Navigation
  - 2.3. Asset Management
    - 2.3.1. Register New Asset
    - 2.3.2. Modify Asset
    - 2.3.3. Approve Asset
    - 2.3.4. Remove/Delete Asset
    - 2.3.5. Categorize Asset
2. Asset Storage and Management (continued)
  - 2.4. Feedback
    - 2.4.1. Collect Comments
    - 2.4.2. Collect Quantitative Feedback
    - 2.4.3. User Registration of Asset Usage
    - 2.4.4. Provider Contact
    - 2.4.5. Display Feedback
  - 2.5. Metrics
    - 2.5.1. Downloads
    - 2.5.2. Links
    - 2.5.3. Registered Users
    - 2.5.4. Rating Summary
  - 2.6. Asset Access Control
    - 2.6.1. Limit Access for Certain Users
3. Notifications
  - 3.1. Send Notifications for Assets
    - 3.1.1. Notify on Modification
    - 3.1.2. Notify on Feedback
  - 3.2. Notification for System Events
    - 3.2.1. Asset Information
    - 3.2.2. System Information
  - 3.3. Notification Management
    - 3.3.1. Add Notifications
    - 3.3.2. Remove Notifications
4. System Operations
  - 4.1. Feedback
    - 4.1.1. System Problems
    - 4.1.2. System Suggestions
    - 4.1.3. Administrator Contact
  - 4.2. Policy Compliant
    - 4.2.1. Verify Provider Info
    - 4.2.2. Verify Provider by Contact
    - 4.2.3. Security, Transmitted Info
    - 4.2.4. Security, Stored Info
    - 4.2.5. Deletion of Users
    - 4.2.6. Protect Private Info
    - 4.2.7. Technical and Other
    - 4.2.8. Policy Availability
  - 4.3. Repository and Catalog
    - 4.3.1. Function as Repository
    - 4.3.2. Function as Catalog
    - 4.3.3. Provider Selects Behavior
    - 4.3.4. Asset Storage Limit
  - 4.4. Asset Cleanup
    - 4.4.1. Deprecation by Content Managers
    - 4.4.2. Removal by Administrators
  - 4.5. Data Integrity
    - 4.5.1. Verification of Data by Providers



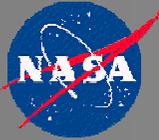
## Architectures Reviewed

- Software packages reviewed:
  - GForge (collaborative development tool based on SourceForge)
  - Savane (Web-based Libre Software hosting system based on SourceForge)
  - XOOPS (Content Management System)
- Other packages inspected, but not reviewed in detail:
  - Fedora Digital Repository System
  - JBoss Portal
  - Liferay Portal
  - Repository in a Box

*In general, these other packages would require large modifications to meet our requirements. They also did not appear to be as simple to create and use as the others, which were selected to represent different types of possible solutions.*

- Available systems reviewed:
  - Global Change Master Directory (GCMD)
- Other systems inspected, but not reviewed in detail:
  - ARC Open Source site
  - GSFC Open source site

*In general, these other systems are basic web pages, and do not have any real catalog/repository software behind them. They also focus on one specific type of asset (NASA open source), preventing them from fully servicing the needs of the community of Earth science software developers.*



## Summary of Reviews

- 
- Requirements matching was performed on all packages/systems.
  - Developers and maintainers were interviewed about:
    - GForge, with the SourceMotel system staff
    - GCMD, with the GCMD staff (done during the Trade Study)
  - By comparison with Savane:
    - GForge was not as flexible or as maintainable.
    - The GCMD would require more effort to develop since the system is not designed for software reuse, and maintenance would be high since it would be a non-standard instance of the system.
  - In addition, the SourceMotel staff indicated that GForge was not suitable for our requirements.
  - Development efforts for Savane and XOOPS, the top two choices, were performed based on estimates of the lines of code required to satisfy requirements and the complexity of the necessary modifications.

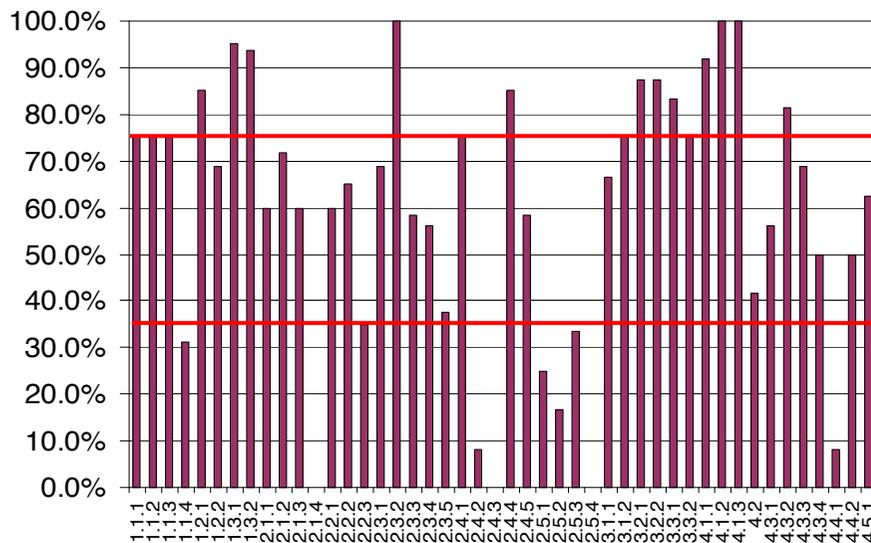


- At the 2006 Joint ESDS Working Group Meeting, requirements satisfaction evaluations were performed by WG members to determine how well selected systems met our requirements.
- Technical evaluations were also performed by the WG to determine if requirements were met, regardless of how well they were implemented.
  - Complexity of each unmet or partially met requirement was estimated, along with the number of lines of code needed to modify the software to meet these requirements.
  - The original Cost Constructive Model (COCOMO) by Barry Boehm was used to estimate development effort for each requirement.



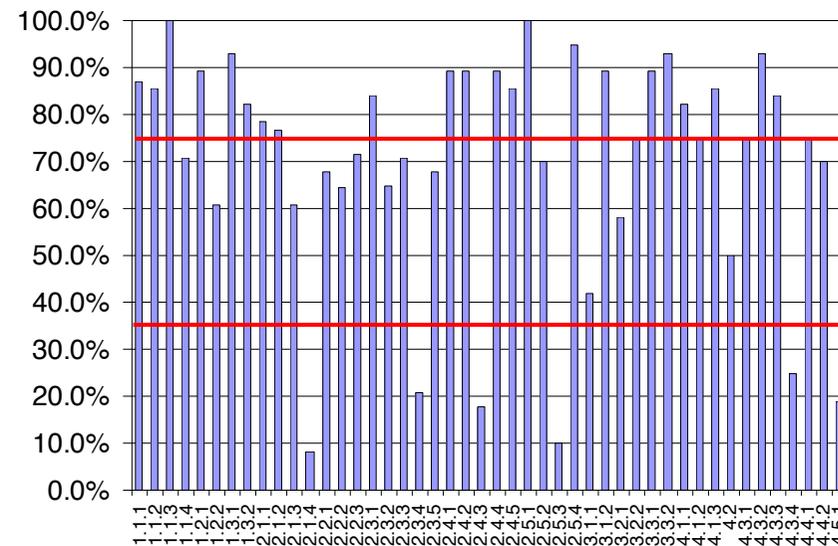
# Requirements Satisfaction

## Savane Requirements Satisfaction



- 18 Requirements Satisfied (75%-100%)
- 18 Requirements Partially Satisfied (35%-75%)
- 10 Requirements Not Satisfied (0%-35%)

## XOOPS Requirements Satisfaction



- 26 Requirements Satisfied (75%-100%)
- 14 Requirements Partially Satisfied (35%-75%)
- 6 Requirements Not Satisfied (0%-35%)

*Note: satisfactions of 0% were rated as 0%; they are not due to lack of responses.*



## Technical Evaluations

<b>Approach Studied</b>	<b># Req's Met</b>	<b># Req's Not Met</b>	<b># Req's Partially Met</b>	<b>Development Effort Estimate [staff-months]</b>
XOOPS	40	9	5	8.12
Savane	24	20	10	34.10
GCMD	26	24	4	N/A
GForge	20	26	8	N/A

*It is important to note that (lack of) satisfaction with the requirements does not necessarily correspond to (not) meeting the requirements.*

See Section 7 of the Reuse Enablement System (RES) Architecture Study document for additional information.

The total number of requirements here (54) is higher than that of the satisfaction evaluations (46) because the latter did not separate out R4.2 into its components and R2.6.1 was added later.

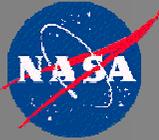


- 
- The requirements satisfaction evaluations are consistent with the technical ability evaluations considering the conditions of the former (e.g., small number statistics).
  - The XOOPS content management system is clearly preferred over the SourceForge-based Savane package in both evaluations.
  - The estimated development effort is 8 months for XOOPS compared to 34 months for Savane.
  - By comparison to Savane:
    - GForge would require at least 34 months, and would be harder to maintain.
    - A GCMD system would require at least 34 months, and be harder to develop since it is not designed for the purposes of software reuse.



## Architecture Study Conclusions

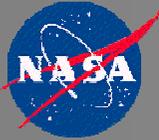
- XOOPS meets more and fails fewer of our requirements than Savane, resulting in a significantly lower estimated development time for XOOPS.
- XOOPS uses modules to provide functionality for the system, and each module is a self-contained component. This makes it easier to modify XOOPS.
- XOOPS is designed for software reuse, while the GCMD is not. Adding the capabilities to support a reuse enablement system is not currently within the scope of the GCMD project.
- *Since XOOPS requires the least amount of development and falls within the scope of the Reuse WG, this open source content management system should be used to create a prototype RES for internal NASA use.*



## Savane Development Effort

Req.	LOC	Complexity	Effort
R1.1.3	550	5	1.54
R1.1.4	1600	9	6.33
R1.2.2	150	5	0.36
R2.1.1	650	6	1.85
R2.1.4	275	10	0.76
R2.2.1	5	3	0.01
R2.2.3	650	6	1.85
R2.3.3	1150	10	4.26
R2.3.5	510	5	1.41
R2.4.1	550	5	1.54
R2.4.2	510	5	1.41
R2.4.3	110	5	0.25
R2.4.4	60	5	0.13
R2.5.1	510	5	1.41
R2.5.2	510	5	1.41
R2.5.3	500	3	1.16
R2.5.4	500	3	1.16

Req.	LOC	Complexity	Effort
R2.6.1	600	5	1.69
R3.1.1	100	3	0.21
R3.1.2	100	3	0.21
R3.2.1	100	3	0.21
R3.2.2	100	3	0.21
R3.3.1	600	5	1.69
R3.3.2	100	3	0.21
R4.2.1	600	6	1.69
R4.2.2	110	5	0.25
R4.2.7	150	3	0.33
R4.2.8	100	2	0.21
R4.3.4	10	4	0.02
R4.4.1	105	5	0.24
<b>Total effort [staff-months]</b>			<b>34.01</b>
<b>Total effort [staff-years]</b>			<b>2.83</b>



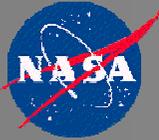
## XOOPS Development Effort

Req.	LOC	Complexity	Effort
R2.1.4	315	4	0.82
R2.2.1	10	2	0.02
R2.3.3	1050	3	2.53
R2.4.3	510	3	1.18
R2.4.4	210	3	0.47
R2.5.3	500	1	1.16
R4.2.2	210	3	0.47
R4.2.3	10	1	0.02
R4.2.7	150	2	0.33
R4.2.8	100	1	0.21
R4.3.1	10	1	0.02
R4.3.3	100	1	0.21
R4.4.1	105	3	0.24
R4.5.1	205	3	0.45
<b>Total effort [staff-months]</b>			<b>8.12</b>
<b>Total effort [staff-years]</b>			<b>0.68</b>



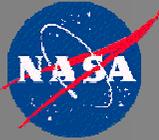
## Example 4: Extensibility Topic Levels

Level	Extensibility Level Summary
1	No ability to extend or modify program behavior
2	Prohibitive costs and efforts need to modify or extend the system
3	Can be extended with the input of considerable time and effort on par with recreating system separately
4	Can be modified and extended through configuration changes, minimal modification of source
5	Consideration for future extensibility designed into system, extensibility approach somewhat defined
6	Designed from the start to allow easy extensibility, provides many points of extensibility and a thorough and detailed extensibility plan
7	Proven to be extensible internally, code structured to provide loose coupling and high cohesion
8	Proven extensibility on a major external program, provides a clear plan for modifying and extending features
9	Proven extensibility in multiple scenarios, provides specific documentation and features to build extensions



## Example 5: Modularity Topic Levels

Level	Modularity Level Summary
1	No designs for modularity or reuse
2	
3	Modularity at major system or subsystem level only
4	
5	Partial segregation of generic and specific functionality
6	
7	Clear delineations of specific and reusable components
8	
9	All functions and data encapsulated into objects or accessible through web service interfaces



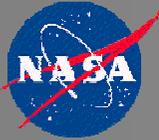
## Example 6: Packaging Topic Levels

Level	Packaging Level Summary
1	Source code available
2	
3	Detailed installation instructions available
4	
5	Software is easily configurable for different environments
6	
7	OS detect and auto-build for supported platforms
8	
9	GUI installation environment provided



## Example 7: Portability Topic Levels

Level	Portability Level Summary
1	The software is not portable at any cost
2	Some parts of the software may be portable
3	The software is only portable with significant costs
4	The software may be portable at a reasonable cost
5	The software is moderately portable
6	The software is portable
7	The software is highly portable
8	
9	The software is completely portable



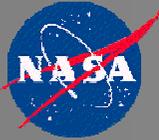
## Example 8: Standards Compliance Topic Levels

Level	Standards Compliance Level Summary
1	Follows no particular standard
2	Follows some parts of common standards and best practices
3	Follows a company-wide standard for development and testing
4	Most components follow a complete, universal standard, but not validated
5	All components follow a universal standard, but only partially validated
6	Validated to follow a specific proprietary standard
7	Validated to comply to a specific open standard
8	Proven by validation to comply with a “gold” standard
9	“Gold” standard compliance of entire system and development, independently validated



## Example 9: Verification/Testing Topic Levels

Level	Verification/Testing Level Summary
1	No testing performed
2	Software application formulated and unit testing performed
3	Testing includes testing for error conditions and proof of handling input errors
4	Software application demonstrated in a laboratory environment
5	Software application tested and validated in a laboratory environment
6	Software application demonstrated in a relevant environment
7	Software application tested and validated in a relevant environment
8	Software application “qualified” through test and demonstration (meets requirements) and successfully delivered to the relevant environment
9	Actual software application tested and validated through successful use of application output



---

Here are links to a number of documents on TRLs and other measures:

- <http://www.hq.nasa.gov/office/codeq/trl/trl.pdf>
- [http://esto.nasa.gov/files/TRL\\_definitions.pdf](http://esto.nasa.gov/files/TRL_definitions.pdf)
- <http://isd.gsfc.nasa.gov/Technology/TRL/TRL.ppt>
- <http://www.stsc.hill.af.mil/crosstalk/2005/05/0505Gold.pdf>
- <http://www.opfro.org/index.html?Components/WorkProducts/ArchitectureSet/TechnologyReadinessAssessment/TechnologyReadinessAssessment.html~Contents>
- <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02sr027.pdf>
- <http://csdl2.computer.org/comp/proceedings/hicss/2005/2268/09/22680315a.pdf> and  
<http://www.sei.cmu.edu/pub/documents/04.reports/pdf/04tr013.pdf>
- <http://www.iccbss.org/2004/proceedings/ImpACT.pdf>
- [http://www.openbrr.org/docs/BRR\\_whitepaper\\_2005RFC1.pdf](http://www.openbrr.org/docs/BRR_whitepaper_2005RFC1.pdf)
- <http://www.hq.nasa.gov/office/codeq/trl/r&d3.pdf>
- <http://www.dtic.mil/ndia/2003systems/nolte.ppt>
- <https://acc.dau.mil/CommunityBrowser.aspx?id=25811>