"RForge" Software Development Site for R packages

DRAFT Revision: 641

Gregory R. Warnes
Associate Director
Groton Nonclinical and Clinical Sciences Statistics
Pfizer Global Research and Development

Date: 2005-08-29 12:03:18 -0400 (Mon, 29 Aug 2005)

1 Introduction

The R statistical software environment is increasingly being used for data analysis within regulated industries (e.g. pharmaceutical research and development). A major attraction of the R system is the relative ease with which add-on packages can be developed and deployed, as well as the extensive set of packages that are already available. (At the time of this writing, more than 400 add-on packages are available from the R-project web site.) As a consequence, add-on packages implementing advanced statistical methods often appear concurrent with–or even before–the publication of the manuscripts describing the methods themselves. The combination of these factors has placed R as a premier tool for statistical computation in many rapidly developing areas, including bioinformatics and systems biology.

The core R environment, under the guidance of the R Core Team, is developed using a standard software model supported by software development tools[1] hosted by the R Project web sites (`http://www.r-project.org`). This makes it a relatively straightforward task to validate the use of R itself for regulatory environments. In

[1]These tools include a source code version control system, an issue-tracking system, file release area, developer and user mailing lists, and web documentation.

fact, Tony Rossini of Novartis is leading an effort to provide the necessary documentation to make this possible.

Unfortunately, the R project does not currently supply equivalent software development tools for authors of user contributed packages. This leaves the authors of these packages to use whatever software development tools they personally have access to.

While the R project requires a contributed package to pass a minimal set of tests that ensure proper package structure and consistency[2], there is no mechanism for categorizing the quality of these add-on packages. As a consequence, the quality of add-on R packages is extremely varied and there is no straightforward mechanism for determining package quality. This makes it extremely difficult to validate the use of add-on R packages for use in a regulated environment.

This project proposes to develop an internet portal that provides add-on package developers with the same types of tools that are available to the developers of the core R environment, and to institute a system for clearly categorizing packages according to their development status. This will encourage package developers to utilize a standard software lifecycle, which will increase the overall quality of contributed R packages, will make it easier to determine which package may be appropriately used in specific circumstances, and will ultimately make it possible to validate a set of well-supported packages for use in regulated environments.

2 Goals

This project has three primary goals:

1. Enable and encourage the use of an appropriate software development lifecycle for user-contributed R add-on packages.

2. Reward package authors with publication credit for the development of high-quality add-on packages.

3. Provide a set of well-designed packages with a documented software lifecycle process that can easily be validated for use in regulated environments.

[2]This set of tests will automatically perform additional regression and functionality tests if appropriate code is in place. However, such code is not required.

3 Benefits

Providing an appropriate set of software lifecycle tools to R packages developers will:

1. Encourage the use of appropriate software lifecycle methods to R packages
2. Allow users to easily evaluate the development status of available packages.
3. Encourage the development of high-quality add on packages
4. Reduce the risk in using add-on R packages
5. Provide a set of packages which can be easily validated for use in regulated environments (GxP, FDA Part 11 compliance, etc.)

4 Mechanism

Create and support an internet portal for add-on package developers that provides software lifecycle tools. These tools will include:

- version control system (subversion)
- issue tracking system
- file release area
- web page
- news lists

for each individual package.

This portal will support categorizing software releases as

- development,
- testing,
- production, or
- peer-reviewed/validated.

Appropriate qualifications for each release classification will be established. Examples of such qualifications include:

- All package releases must pass the 'R CMD check' standard R package tests.
- Test packages must include working examples for each documented function. These will be used for basic regression testing.
- Production packages must include a 'vignette' describing the basic features of the package and show how these features are used in a real analysis.
- Peer-reviewed/validated packages must include a reasonably complete set of unit tests, and must be subjected to code review by two independent individuals.

In order to encourage academic package developers to submit their packages to the peer-review/validation process, packages that are accepted into this category will become publications in a well recognized peer-reviewed journal, probably the *Journal of Statistical Software* (http://jstatsoft.org).

In order to obtain a sufficient pool of software reviewers, each individual submitting a package to the peer review process will be required to commit to reviewing (or arranging for reviews) of two other software packages. Alternatively, individuals may elect to pay a fee which will be used to remunerate reviewers for timely effort.

Software reviewers will complete a standard review form developed by the community which will ensure that best practices are followed by both software authors and by reviewers. This form will include a commentary on the software that will be published alongside the software itself once all revisions have been completed and the package has been accepted for publication. This commentary will appear under the reviewers names, mirroring standard software reviews, ensuring that package reviewers also receive appropriate publication credit for their work.

5 Resources

5.1 Initial Site Creation

- Initial Design Specification – 1 man month
- Programming – 3 man months

  This process should be a straightforward extension of existing sourceforge-style tools. One particularly attractive option is 'GForge', available from http://gforge.org/ and as a Debian package.

  This will include

  - installation and configuration of the web server
  - design of the web interface and project templates
  - import of existing R packages
  - integration with the existing R project web site & tools
  - security evaluation and system hardening

- User Acceptance Testing – 3 months?
- Train site maintainer – 3 month? (concurrent with acceptance testing?)

5.2 Ongoing Maintenance

- Web site hosting ≈$500/yr (guess)
- Web site administrator and maintainer – 20hrs/week (probably an over-estimate)

  This individual will:

  1. Ensure the ongoing operation of the web site, include host and site administration (regular backups!).
  2. Provide technical support for package developers.
  3. Perform some ongoing development to improved the available tools, site documentation, category standards, etc.
  4. Acting as a gatekeeper for category changes, including assignment of reviewers, processing of review responses, etc.

5.3 Total Resources

Under the preliminary resource estimates, funding a total of 1 year of full-time-equivalent (FTE) in salary and benefits, plus a small additional cost for equipment and web hosting (<$20K?), should be sufficient to establish and maintain the R package development portal for 1 full year. Future years should require only 1/2 FTE plus web hosting costs (≈$500?).

6 Funding

We propose that interested organizations (Pfizer, Insightful, Novartis, etc.) jointly fund the R Foundation (http://www.r-project.org/foundation) to perform the work required to create and then maintain the R package development site.

There are a number of reasons for using the R Foundation to perform this work. First and most importantly, the R Community must have confidence in the organization developing and maintaining the package development portal. As the R core and key R developers are the primary members of the R Foundation, this ensures that they are involved in oversight of the development and maintenance of the portal.

Second, funding the project through the R Foundation provides a straightforward mechanism for a multiple organizations to participate, potentially reducing the cost for individual participants, while creating a larger total funding pool.

Third, the R Foundation is a logical organization to shepherd the continued operation of the R package development portal over the long term. The continued operational costs can be funded through annual contributions from a variety of sources, potentially including government grants.

Fourth, as an Austrian not-for-profit corporation, contributions are tax-deductible for European organizations. With an appropriate filing (see http://www.r-project.org/foundation/donations.html), this can also be arranged for US organizations.

Finally, the R community would like to see the R Foundation serve as a resource accumulator for funding a variety of enchantments to the R software system that are not easily accomplished via the current contributed time model. Funding the R package development portal via the R foundation will demonstrate the utility of this mechanism.