Template for Grading Scientific Computing Software

Spencer Smith, Adam Lazzarato, Yue Sun, Zheng Zeng,
Vasudha Kapil and Jacques Carette

April 29, 2020

The table below lists the full set of measures that are assessed for each software product. The measures are grouped under headings for each quality, and one for summary information. Following each measure, the type for a valid result is given in brackets. Many of the types are given as enumerated sets. For instance, the response on many of the questions is one of "yes," "no," or "unclear." The type "number" means natural number, a positive integer. The types for date and url are not explicitly defined, but they are what one would expect from their names. In some cases the response for a given question is not necessarily limited to one answer, such as the question on what platforms are supported by the software product. Case like this are indicated by "set of" preceding the type of an individual answer. The type in these cases are then the power set of the individual response type. In some cases a superscript * is used to indicate that a response of this type should be accompanied by explanatory text. For instance, if problems were caused by uninstall, the reviewer should note what problems were caused.

Table 1: Grading Template

Summary Information
Software name? (string)
URL? (url)
Educational institution (string)
Software purpose (string)
Number of developers (number)
How is the project funded (string)
Number of downloads for current version (number)
Release date (date)
Last updated (date)
Status ({alive, dead, unclear})
License ({GNU GPL, BSD, MIT, terms of use, trial, none, unclear})
Platforms (set of {Windows, Linux, OS X, Android, Other OS})
Category ({concept, public, private})
Development model ({open source, freeware, commercial})
Publications using the software (set of url)
Publications about the software (set of url)
Is source code available? ({yes, no})
Programming language(s) (set of {FORTRAN, Matlab, C, C++, Java, R, Ruby, Python, Cython, BASIC, Pascal, IDL, unclear})
Installability (Measured via installation on a virtual machine.)
Are there installation instructions? ({yes, no})
Are the installation instructions linear? ({yes, no, n/a})
Is there something in place to automate the installation (makefile, script, installer, etc)? ({yes*, no})
Is there a specified way to validate the installation, such as a test suite? ({yes*, no})

How many steps were involved in the installation? (number)
How many software packages need to be installed before or during installation? (number)
Run uninstall, if available. Were any obvious problems caused? ({unavail, yes*, no})
Overall impression? ({1 .. 10})

Correctness and Verifiability

Are external libraries used? ({yes*, no, unclear})
Does the community have confidence in this library? ({yes, no, unclear})
Any reference to the requirements specifications of the program? ({yes*, no, unclear})
What tools or techniques are used to build confidence of correctness? (string)
If there is a getting started tutorial, is the output as expected? ({yes, no*, n/a})
Overall impression? ({1 .. 10})

Surface Reliability

Did the software "break" during installation? ({yes*, no})
Did the software "break" during the initial tutorial testing? ({yes*, no, n/a})
Overall impression? ({1 .. 10})

Surface Robustness

Does the software handle unexpected/unanticipated input (like data of the wrong type, empty input, missing files or links) reasonably? (a reasonable response can include an appropriate error message.) ({yes, no*})
For any plain text input files, if all new lines are replaced with new lines and carriage returns, will the software handle this gracefully? ({yes, no*, n/a})
Overall impression? ({1 .. 10})

Surface Performance

Is there evidence that performance was considered? ({yes*, no})
Overall impression? ({1 .. 10})

Surface Usability

Is there a getting started tutorial? ({yes, no})
Is there a standard example that is explained? ({yes, no})
Is there a user manual? ({yes, no})
Does the application have the usual "look and feel" for the platform it is on? ({yes, no*})
Are there any features that show a lack of visibility? ({yes, no*})
Are expected user characteristics documented? ({yes, no})
What is the user support model? (string)
Overall impression? ({1 .. 10})

Maintainability

Is there a history of multiple versions of the software? ({yes, no, unclear})
Is there any information on how code is reviewed, or how to contribute? ({yes*, no})

Is there a changelog? ({yes, no})
What is the maintenance type? (set of {corrective, adaptive, perfective, unclear})
What issue tracking tool is employed? (set of {Trac, JIRA, Redmine, e-mail, discussion board, sourceforge, google code, git, none, unclear})
Are the majority of identified bugs fixed? ({yes, no*, unclear})
Which version control system is in use? ({svn, cvs, git, github, unclear})
Is there evidence that maintainability was considered in the design? ({yes*, no})
Are there code clones? ({yes*, no, unclear})
Overall impression? ({1 .. 10})

Reusability

Are any portions of the software used by another package? ({yes*, no})
Is there evidence that reusability was considered in the design? (API documented, web service, command line tools, ...) ({yes*, no, unclear})
Overall impression? ({1 .. 10})

Portability

What platforms is the software advertised to work on? (set of {Windows, Linux, OS X, Android, Other OS})
Are special steps taken in the source code to handle portability? ({yes*, no, n/a})
Is portability explicitly identified as NOT being important? ({yes, no})
Convincing evidence that portability has been achieved? ({yes*, no})
Overall impression? ({1 .. 10})

Surface Understandability (Based on 10 random source files)

Consistent indentation and formatting style? ({yes, no, n/a})
Explicit identification of a coding standard? ({yes*, no, n/a})
Are the code identifiers consistent, distinctive, and meaningful? ({yes, no*, n/a})
Are constants (other than 0 and 1) hard coded into the program? ({yes, no*, n/a})
Comments are clear, indicate what is being done, not how? ({yes, no*, n/a})
Is the name/URL of any algorithms used mentioned? ({yes, no*, n/a})
Parameters are in the same order for all functions? ({yes, no*, n/a})
Is code modularized? ({yes, no*, n/a})
Descriptive names of source code files? ({yes, no*, n/a})
Is a design document provided? ({yes*, no, n/a})
Overall impression? ({1 .. 10})

Interoperability

Does the software interoperate with external systems? ({yes*, no})
Is there a workflow that uses other softwares? ({yes*, no})
If there are external interactions, is the API clearly defined? ({yes*, no, n/a})
Overall impression? ({1 .. 10})

Visibility/Transparency
Is the development process defined? If yes, what process is used. ({yes*, no, n/a})
Ease of external examination relative to other products considered? ({1 .. 10})
Overall impression? ({1 .. 10})
Reproducibility
Is there a record of the environment used for their development and testing? ({yes*, no})
Is test data available for verification? ({yes, no})
Are automated tools used to capture experimental context? ({yes*, no})
Overall impression? ({1 .. 10})