

RSICC Quality Assurance (QA)

What is QA?? In general, the objective of QA is to **supply a consistent value (within fixed defined limits) of a product and/or operation**. Value is defined by the user/customer and scientist/developer. Neither a QA'd product alone nor a QA use of a product guarantees valued results. The QA process prescribes the development process and the product's use, increasing the probability of valued results.

Software/data author generated QA:

Documentation: Software and data QA defines a level of quality desired in the output (either as a model of a measurable quantity or as a predictive outcome of events) which is not an inherent characteristic of the outcome. **QA establishes a high level of certainty** that the value (model accuracy, prediction precision) of a software product is within defined fixed limits. The documentation process is an important step in generating a cohesive software/data 'package' that will ultimately be useful to the user or a future author's revision improving the software's value. Documentation includes theory and background information, motivation or problem solved, components defining the problem(s) for valued solutions, data sets to be used, the operating system platform(s)¹, software used (compilers, linkers, graphics), and hardware platform(s).

Testing: The testing of software under development and post-development lends credibility to the value of the software. Here the defined limits of use should be explored, noted, and documented. Regular use of 'real' or experimental data, called 'benchmarks', can be extremely desirable. Testing establishes the errors of the software-generated data versus the benchmark solution sets. The idea is to generate a working volume of problems that will fall within the error limits defined by the testing and the benchmarks. When the output of a software program is within these defined accuracy and precision limits; this is called validation.

The ANS (American Nuclear Society) 10 standards subcommittee, chaired by former RSICC director Betty Maskewicz, revised the ANSI/ANS-10.3 standard and is an important guide for interchanging computer programs from one installation to another. It is one of three documents directed toward individuals who develop computer programs. The other two are American National Standard Recommended Programming Practices to Facilitate the Portability of Scientific Computer Programs, ANSI/Am-10.2-1982, and American National Standard Guidelines for Considering User Needs in Computer Program Development, ANSI/ANS-10.5-1979.

RSICC originated QA:

Before RSICC can generate a software package to distribute to the public, many software QA checks are performed. Outlined in 'RADIATION SHIELDING INFORMATION CENTER QUALITY ASSURANCE PLAN' QAP-X-89-EPMIRSIC-001, Version I. I, May 1992, the RSICC QA plan (formerly Radiation Shielding Information Center) details the procedures its staff performs before a software program becomes a package. Essentially mirroring the work of a QA produced software program, proper documentation and testing must have been performed. Each software program and its subprograms are checked against all the author supplied sample problems, usually two, three, up to 20 or more are executed. Using the author designated (and documented) computer hardware and software platforms, sample problem execution output must match that of the author's supplied output data files. The sample problems must run on the given (documented) configuration (software, hardware platform) and match output data exactly; this is called verification. Many different RSICC computers have been used to perform these tests. The Unix, PC, and other hardware platforms have provided the necessary equipment. Software compilers for FORTRAN, C, BASIC, and other languages have been used to test source code provided by the software authors. Once compiled and linked, the programs are run using the sample input files provided by the authors. Differences or difficulties are noted, and included in the RSICC

¹ Includes date and version of software, with any updates or revisions.

generated readme.txt notes sent with the software. Tests that yield differences are resolved with the help of the authors, requiring updated notes, for version control. Changes to source code are made with the cooperation of the authors and RSICC software test staff. The changes usually are those that impact installation problems, data file read errors, printing or output file generation errors, and compiler-specific errors. Sample input files are checked against the originals before packaging and release to the public.

Summary

RSICC staff performs a 'first-order' QA check on the software, looking into areas of program failure or weaknesses that provide a large 'value-added' before the software or data is considered a package ready for release. Further, users of RSICC programs will generate questions that RSICC staff note and assumes the responsibility to acquire their answers. Sometimes external advisers must be used to help, authors, or other members of the software/data development team.

RSICC is ultimately the center for resolving problems that arise in the use of a software program or data and could result in 1) retrieving defective software and notifying customers of the problems, 2) generating a public notice in a newsletter 3) notes in a www base electronic 'notebook' for the specific software, 4) issuing a download of a revision that eradicates the problem, and 5) supply new details about operational validation and benchmarking.