June 1999 • NREL/TP-500-26207

Technical Report

Software Quality-Control Guidelines for Codes Developed for the NWTC

Marshall L. Buhl, Jr.
H. James Green

NREL

National Renewable Energy Laboratory

1617 Cole Boulevard
Golden, Colorado 80401-3393

NREL is a U.S. Department of Energy Laboratory
Operated by Midwest Research Institute • Battelle • Bechtel

Contract No. DE-AC36-98-GO10337

June 1999 • NREL/TP-500-26207

Software Quality-Control Guidelines for Codes Developed for the NWTC

Marshall L. Buhl, Jr.
H. James Green

Prepared under Task No. WE901210

National Renewable Energy Laboratory
1617 Cole Boulevard
Golden, Colorado 80401-3393

NREL is a U.S. Department of Energy Laboratory
Operated by Midwest Research Institute • Battelle • Bechtel

Contract No. DE-AC36-98-GO10337

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Available to DOE and DOE contractors from:
Office of Scientific and Technical Information (OSTI)
P.O. Box 62
Oak Ridge, TN 37831
Prices available by calling 423-576-8401

Available to the public from:
National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
703-605-6000 or 800-553-6847
or
DOE Information Bridge
http://www.doe.gov/bridge/home.html

♻ Printed on paper containing at least 50% wastepaper, including 20% postconsumer waste

Software Quality Control Guidelines for Codes Developed for the NWTC

INTRODUCTION

Members in the wind-energy research, development, deployment, and production communities use computer codes for many things. They base important decisions on the results from the codes. It is important that the developers of these codes scrutinize them to assure an appropriate level for quality.

The National Wind Technology Center (NWTC) and its subcontractors have developed many computer codes now in use in the United States and around the world. This document will present some guidelines for ensuring the quality of programs that are developed for the NWTC.

Not all programs need the same level of quality control (QC). For instance, a code that will have a wide distribution needs a more rigorous QC program than one that only its author will use. There is a wide spectrum between the two. Those responsible for the development of the program must decide how much effort should be devoted to QC.

PURPOSE

First, we simply hope this document will convince software developers to think about quality. We believe they should keep it in mind from the very beginning of their development effort.

Task planners and technical monitors for development subcontracts may want to use these guidelines as a starting point for generating QC requirements. These requirements should specify deliverables and the requirements for keeping records. We leave it to the planners and monitors to decide the appropriate level of effort that should be devoted to software QC.

NOMENCLATURE

Some of the words used in software QC have ambiguous meanings. In order to reduce the chance of miscommunication, a list of terms used in this document and elsewhere appears in Table 1.

Table 1. Software QC Nomenclature

Verifying	Comparing code predictions to hand calculations or other codes.
Validating	Comparing code predictions to test data.
Version Testing	Comparing output from a new version to output from previous versions.
Installation Testing	Comparing output from a newly installed code to sample output distributed with the code.
Development Code	Code tested only by the developer.
Alpha Code	Code tested only internally and possibly externally by a very small number of selected users.
Beta Code	Code tested by a number of users and distributed externally to a modest audience. Most distributed NWTC codes are in this category.
Released Code	Code that is well tested and documented and is officially distributed by DOE's Energy Science and Technology Software Center (ESTSC).

GUIDELINES

Version Tracking

Any code used by more than just the developer should have some form of version tracking. Not only should the developer assign a number to each version of a program, but the program should also display that information when it runs, and it should include the information in output files. We believe it is also appropriate to track version dates too.

One possible numbering scheme for programs is to use a number in the form X.XX, where the one's position changes when the program undergoes a major revision. The tenth's position changes when the code has minor enhancements. The hundredth's position changes for bug fixes and trivial changes.

For codes that produce output that is read by programs controlled by others, it may not be possible to include version information in the output files. Binary output files may also have a similar problem.

Change Log

It is important to keep a log of changes to a program. For each revision, the log should include the name of the person(s) making the change(s), the date, the version number, and a description of the change(s). Descriptions should be brief and should merely list the new features or fixes. It is not necessary to go into details about the development of the first working version of the program.

The log should not appear within the source code, but in a separate file. This allows users to easily review the changes to determine if it is necessary to redo any runs. It may be that commenting changes within the source code is appropriate, but the primary log should be in a separate file.

Archives

It is advisable that developers use an archiving program to create a separate archive file for every version of any program. Archive files should include the executable program, the source files, a read-me file, and sample input and output files. It is useful to include the number of the version in the archive file name.

Verification

All programs should have their results verified. The complexity of the program and the level of distribution should dictate the amount of effort required. Compare the results of the program to hand calculations or the results from other programs. Spreadsheets can be especially useful for this. We think it is appropriate to document and publish the comparisons for widely distributed codes.

Validation

Validation is a difficult subject. It is subjective, probably expensive, and not always possible. Yet, it can add value to a predictive code. It generally does not apply to pre- and post-processing utilities or data-acquisition software.

When validating a code against test data, the answers will never agree. Errors in the code, simplifications in the model, the use of inaccurate model properties, and even errors in the test data can cause discrepancies. Deciding how close is good enough is a matter of personal opinion. Again, the complexity of the program and the level of distribution should determine the necessary degree of accuracy, the number of cases to compare, and the level of documentation.

Version/Installation Checking

Sometimes a change in one part of a program can have unexpected side effects on other parts of the program. Developers should compare results for new versions to results from previous versions. Doing this for a variety of cases reduces the chance of such errors going unnoticed.

For many programs, it is possible to create computer scripts that exercise many of the features of the program. One useful technique is to compare the results from a series of runs with saved versions of the output. Use of file-comparison utilities can automate the documentation of the differences. If distributed with the software, these scripts also allow end users to test the installation of the software.

For programs with graphical user interfaces, computer scripts may not be possible. If the program has the ability to record macros, then it may be possible. If not, developers should document a set of exercises that they repeat for each new version of their programs. The use of file-comparison utilities may still be of some benefit here.

Documentation

For many codes, read-me files, on-line help, or well-commented input files are usually sufficient for documentation. More-complex codes probably need a well-written user's manual. Widely distributed codes or those used for turbine certification should also have the theory, verification, and validation fully documented.

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
June 1999

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Software Quality-Control Guidelines for Codes Developed for the NWTC

5. FUNDING NUMBERS
WE901210

6. AUTHOR(S)
Marshall L. Buhl Jr., H. James Green

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TP-500-26207

11. SUPPLEMENTARY NOTES

NREL Technical Monitor:

12a. DISTRIBUTION/AVAILABILITY STATEMENT
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Members in the wind-energy research, development, deployment, and production communities use computer codes for many things. They base important decisions on the results from the codes. It is important that the developers of these codes scrutinize them to assure an appropriate level for quality.

The National Wind Technology Center (NWTC) and its subcontractors have developed many computer codes now in use in the United States and around the world. This document will present some guidelines for ensuring the quality of programs that are developed for the NWTC.

14. SUBJECT TERMS
Software, Quality Control, Guidelines

15. NUMBER OF PAGES
2

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT