SCALE-CMP-004, Rev. 5

SCALE Procedure for Discrepancy Reports

Prepared by
B. T. Rearden, J. P. Lefebvre, and S. M. Bowman

Reactor and Nuclear Systems Division (RNSD)
Oak Ridge National Laboratory

Date Revised: May 22, 2013

Approvals:

Bradley T. Rearden

Digitally signed by Bradley T.
Rearden (qol)
Date: 2013.05.22 15:47:13 -04'00'

May 22, 2013

SCALE Project Leader — Date

Cecil V. Parks

June 5, 2013

Director, Reactor and Nuclear Systems Division — Date

Matthew T. Sieger

Digitally signed by Matthew T. Sieger
DN: cn=Matthew T. Sieger, o=Oak Ridge National
Laboratory, ou=Performance Analysis and Quality (PAQ),
email=siegermt@ornl.gov, c=US
Date: 2013.05.24 15:50:38 -04'00'

May 24, 2013

Quality Manager, Performance Analysis and Quality — Date

SCALE Procedure for Discrepancy Reports

1.0 PURPOSE

To describe a standardized operational procedure to control the use of the SCALE Discrepancy Report (SDR) to (1) identify a situation where the SCALE code system fails to perform according to the documentation, (2) indicate the impact to current and past users, and (3) recommend action to resolve and/or temporarily circumvent the discrepancy. The procedure outlined in this document complies with the Configuration Management Plan (CMP) for the SCALE code system, which is controlled by the Reactor and Nuclear Systems Division (RNSD).

2.0 SCOPE

A SCALE Discrepancy Report should be filed whenever any SCALE program or data library included in an external distribution fails to perform according to its software documentation. This includes documentation errors.

3.0 DEFINITIONS

Discrepancy - The failure of software or data to perform according to its documentation. This may be due to an error in the software, data, or documentation. An operational failure or incorrect results are both examples of software discrepancies.

4.0 PROCEDURE

4.1 Problems encountered with SCALE are reported to the Project Leader and recorded in an electronic SCALE Quality Assurance Bug Case. The Bug Case includes the following information:

- *Steps to reproduce* – A complete description of what is needed to encounter the discrepancy including SCALE version and platform as well as any necessary input files or data
- *What you expected to see* – Description of the expected result if the bug were not encountered
- *What you saw instead* – Description of result that is believed to be discrepant

The Bug Case status is set to Active (Initial report).

4.2 The Project Leader reviews the report, enters the following information, modifies the status to Active (In Review) and assigns the Case to a staff member for assessment.

- Charge Number – ORNL billing number to be used for work related to this issue
- Sponsor – Name of sponsor for this work
- Technical Reviewer – Staff member who will review the assessment of this issue
- Milestone – SCALE version where this Bug was first observed
- Estimate – Amount of labor effort in units of time that is anticipated to investigate this issue

4.3 The assigned staff member assesses the information provided in the Bug Case, requests any necessary additional information, annotates the Bug Case with an assessment, and assigns the Case to the Project Leader for review.

4.4 If the issue is confirmed as a discrepancy, the Bug Case status is updated to Active. The report is assigned a SCALE Discrepancy Log (SDL) identifier and docketed by the SQA Coordinator. The docket number begins with SDL and is assigned a sequence number in the form *SDL-YYYY-NNN*. *YYYY* equals the current year and *NNN* equals a number beginning with *001* and increasing by one each time an SDL identifier is assigned. The complete sequence of numbering restarts at the beginning of each fiscal year.

If the issue is identified as expected behavior, an SDL identifier is not assigned, and the issue is marked as Resolved per 4.8 below.

4.5 The Project Leader assesses the issue to determine if it is a *Significant Software Error*, gathering supporting information from SCALE staff as necessary to form an accurate assessment.

Significant Software Errors are defined here as those program or data errors that occur with no warning or error messages, appear to allow proper execution of the software yet provide results that are:

- Inconsistent with the evaluated nuclear data or the theory models applied in the codes, and
- Judged to be of potential significance to operational safety (e.g., potential k_{eff} error greater than 1%).

Items believed to be *Significant Software Errors* are reported to the Leadership Team for review. If the Leadership Team confirms the assessment, the discrepancy is reported to the Director of RNSD for final review. If the Director of RNSD confirms the assessment, *Significant Software Errors* are reported in accordance with 0 below and include the information listed in 4.12.

4.6 The Bug Case is assigned to a Developer for corrective action. The corrective actions are logged in the Bug Case, including but not limited to code, data, and/or documentation updates, which are tracked in an additional SCALE Quality Assurance Feature Case (SCALE-CMP-013) that is cross referenced with the Bug Case. The Developer assigns the Bug Case to the designated Technical Reviewer.

4.7 The Technical Reviewer assesses the issue and any corrective action, coordinates any revisions with the Developer, enters notes regarding the review, and assigns the Case to the Project Leader.

4.8 The Project Leader approves the corrective action (if any), enters QA comments to succinctly describe the issue, and changes the Bug Case status from Active to Resolved. The following resolution statuses may be used:

- Resolved (Fixed) – The issue was a discrepancy and appropriate corrective actions have been implemented.
- Resolved (Not Reproducible) – The issue could not be reproduced and is not tracked as a discrepancy.
- Resolved (Duplicate) – The issue was previously reported in another Bug Case, which is cross referenced as the current Case is resolved.
- Resolved (Postponed) – The issue will be resolved with other planned revisions. Notifications are issued as deemed appropriate by the Project Leader.
- Resolved (Won’t Fix) – A potentially undesirable feature is performing as expected and will not be corrected. These features may be subject to review in future development. Notifications are issued as deemed appropriate by the Project Leader.
- Resolved (By Design) – The feature is performing as designed with no corrective action required.

4.9 The Project Leader ensures that RSICC, users, and sponsors are notified of the corrective action and its potential impact on users via e-mail notices, web postings, and/or the SCALE Newsletter as deemed appropriate.

4.10 The SDL docket is updated by the SQA Coordinator.

4.11 Contacts and Expectations for Notification of *Safety-Significant Software Errors*

Organization/Group	Points of Contact/Expectations
DOE	• **Nuclear Criticality Safety Program Manager** • **Packaging Certification Program Manager** • **Nuclear Fuels Storage and Transportation Planning Project National Technical Director** Interacts with SCALE Project Leader to understand error, judge impact on operational safety, review checklist, and make decision on issuing as *Significant Software Error*. Coordinates notification issuance for any errors deemed *significant* and interacts with DOE offices, DOE facilities, and other government organizations.
NRC	• **Office of Nuclear Material Safety and Safeguards SCALE Project Manger** • **Office of Nuclear Regulatory Research SCALE Project Manager** Interacts with SCALE Project Leader to understand error, judge impact on operational safety, review checklist, and make decision on issuing as *Significant Software Error*. Coordinates notification issuance for any errors deemed *significant* and interacts with NRC offices, licensees, and other government organizations.
RSICC	Provides notice in RSICC Newsletter and via e-mail alert to recipients of the code version(s) affected.
Code Developers	**SCALE Project Leader –** Prepares and reviews checklist. Interacts with DOE, NRC, and RSICC. Issues the notification to any user groups pertinent to the software. Notification of errors (or discrepancies where code does not perform as described by documentation) that are not deemed to be *significant* may use this checklist format as deemed appropriate. Notifies those on the *SCALE News* email list (scalenews@home.ornl.gov) and assures the information is posted in the SCALE website.

4.12 Checklist for Significant Software Error Notification

Item	Description
Software Identification	System/code name and version(s) impacted, including RSICC package identifier (and a discussion of versions NOT impacted may need to be clarified)
Data Library	Comment on whether error is related to use of particular data or data library
Computing platform (Unix, Windows, Linux, etc.)	Comment on whether error is generic to all computing platforms or is platform/compiler-dependent
Description of the error	Simple explanation of the data or code error
How was the error identified?	History of how the error was identified and confirmed
When does the error occur?	A description of the situations that would lead to manifestation of the error in a particular user problem. In the case of a program, this description should include the types of problems, geometry modeling characteristics, and/or combination of user input that “activates” the error. In the case of data, the particular data library(ies) that contains the error together with the nuclides and reactions affected by the erroneous data should be identified.
Potential impact of error	Specific quantitative information for specific example problems together with expert judgment from the developers and applications experts should be provided. The expert judgment is needed particularly for errors that could have far-reaching impacts as to the classes of problems that might be impacted. This discussion should be done in the context of users performing analyses using appropriate validation techniques.
Frequency / likelihood of error occurring	A subjective discussion that will depend on 1) the particular combination of input parameters or code options that cause the error to manifest itself in a problem or

	2) the importance of the erroneous data to various code results.
How can users determine if error affects their calculations?	What tests can be made? Is there a tool available to search input files for the combination of input that activates the error?
What action should users take if error affects them?	Can users independently correct their data or input to assure the error is not manifested? What is the most efficient and effective way for users to correct identified cases to assure the error is removed?
Is correction to code/data available?	Are code or data library updates available to correct the error without having the user change existing data sets?
How to obtain/install correction	Information on how to obtain code or library updates.