VERA Virtual Environment for Reactor Applications

ORNL/SPR-2021/2328

MPACT Software Management Plan Version 4.3

June 14, 2022

Thomas Downar[1], Brendan Kochunas[1], Aaron Graham[2], Shane Stimpson[2], and Benjamin Collins[2]

[1]University of Michigan
[2]Oak Ridge National Laboratory

Approved for public release.
Distribution is unlimited.

Oak Ridge National Laboratory
U.S. Department of Energy

ORNL is managed by UT-Battelle LLC for the US Department of Energy.

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.

ORNL/SPR-2021/2328

MPACT SOFTWARE MANAGEMENT PLAN VERSION 4.3

Thomas Downar[1], Brendan Kochunas[1], Aaron Graham[2], Shane Stimpson[2], and Benjamin Collins[2]

[1]University of Michigan
[2]Oak Ridge National Laboratory

Date Published: June 14, 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

MPACT Software Management Plan Version 4.3

Revision Log

Revision	Date	Affected Pages	Revision Description
0	06/14/2022	iv	Updated Program Technical Leader
		n/a	Removed List of Figures and Tables
		8,18	Removed references to PHI Integration Team; corrected typos in Programmer's and User's Manual
		18	Clarified location of MPACT Style Guide
		all	Updated document to VERA template and generated new document number; this supersedes previous document version CASL-U-2019-1872-002

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

MPACT Software Management Plan Version 4.3

Approvals:

Aaron Graham Digitally signed by Aaron Graham
Date: 2022.06.21 13:26:19 -04'00'

Aaron Graham, MPACT Product Software Manager **Date**

Erik Walker Digitally signed by Erik Walker
Date: 2022.06.21 14:28:29 -04'00'

Erik Walker, Independent Reviewer **Date**

Aaron Graham Digitally signed by Aaron Graham
Date: 2022.06.21 13:26:33 -04'00'

Aaron Graham, Program Technical Leader **Date**

Tristan Grover, VERA Software Quality Coordinator **Date**

CONTENTS

ABBREVIATIONS

CMFD	coarse mesh finite difference
ESSM	embedded self-shielding method
MOC	method of characteristics
NEM	nodal expansion method
ORNL	Oak Ridge National Laboratory
PSM	product software manager
QA	quality assurance
QAPP	quality assurance program plan
SBMS	Standards Based Management System
SHA-1	secure hash algorithm 1
SQA	software quality assurance
SQAP	software quality assurance program
UM	University of Michigan
V&V	validation and verification
VSI	VERA Software Integrator

1. PROJECT BACKGROUND AND OBJECTIVES

The MPACT code solves a discretized form of the Boltzmann transport equation on a wide variety of geometries and is distributed with a multigroup neutron cross section library. MPACT provides an advanced geometrically resolved neutral-particle transport capability to solve the flux distribution throughout the entire problem geometry, and it can model the isotopic depletion, decay, and activation of materials. The flux solution in MPACT is provided using a 2D/1D synthesis method within the framework of the 3D coarse mesh finite difference (CMFD) method for which axial and radial correction factors are obtained from 2D method of characteristics (MOC) and 1D nodal expansion method (NEM), PN, or SN. Other key characteristics of the MPACT code include the subgroup method and the embedded self-shielding method (ESSM) for resonance treatment, depletion capability based on the ORIGEN exponential matrix method, and a simplified thermal-hydraulics method for temperature/fluid feedback. The sole purpose of the simplified feedback model is to provide a mechanism for testing during code development and to provide a limited capability for educational applications.

Work performed at the code level supports the VERA-QA-001, quality assurance program plan (QAPP) and VERA-QA-002, VERA Software Quality Assurance Plan.

2. PLAN DESCRIPTION

The MPACT code team is responsible for implementing software engineering and software quality assurance (SQA) requirements under its control as defined in this plan. Compliance with this plan is required throughout the software life cycle, including planning, requirements, acquisition, design, implementation, acceptance testing, maintenance and operations, and retirement.

The purpose of this document is to define the MPACT processes that are applicable to the VERA project. MPACT is a stand-alone code, and multiple organizations are associated with MPACT development. However, Oak Ridge National Laboratory (ORNL) and the University of Michigan (UM) will control software as defined within this document such that it properly interacts and functions within the VERA product suite.

The scope of application for this plan is the “MPACT Software Configuration Item List” which is described in the “Software Configuration Management” section of this plan. Software that does not affect the performance of the VERA product suite is not within the scope of this plan.

2.1 ASSUMPTIONS AND CONSTRAINTS

- Adequate funding, required hardware, and system software are available to complete the planned MPACT activities.
- Responsibilities can be delegated for a temporary period to alternate personnel; however, the personnel designated in this plan retain ultimate responsibility. A designee must have the appropriate knowledge, required training, and required independence as identified in the “Training” section of this plan.
- The hardware that serves MPACT applications is managed by the Infrastructure team. Management of this hardware is outside the scope of this plan.

3. REFERENCES

The following requirements documents apply to development of this plan:

VERA-QA-001, VERA Quality Assurance Program Plan (QAPP)

VERA-QA-002, VERA Software Quality Assurance Plan (SQAP)

VERA-QA-003, VERA Management of Controlled Documents and Records

VERA-QA-004, VERA Software Configuration and Control

VERA-QA-005, VERA User's Manual and Theory Manual

VERA-QA-006, VERA Requirements and Design Process

VERA-QA-009, VERA Release Process

VERA-QA-010, VERA Problem Reporting and Corrective Action

4. DEFINITIONS AND ACRONYMS

The following section defines the terms used in this plan. Acronyms and abbreviations are defined at their first use in this plan. Defined terms and document titles are indicated in this plan by italic font. Plan sections or appendix titles are set in quotation marks.

baseline: a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for use and further development, and that can be changed only by using an approved change control process. Source: ASME NQA-1a-2009

branch (or *topic branch*): an offshoot from the MPACT Git repository that is not the master branch. More information can be found at `https://guides.github.com`.

code repository (alternately, *MPACT repository*, *MPACT code repository*, or simply *repository*): archive that stores all source code, tests, configuration instructions, and documentation. Version control is performed using the Git version control system (see `https://guides.github.com`).

commit: a specific configuration of the code repository that is identified by a unique secure hash algorithm 1 (SHA-1) number. See `https://guides.github.com` for more information.

Gitlab: internal ORNL and UM website for hosting the MPACT code repository, running automated continuous integration tests, and submitting and performing merge request reviews. *Gitlab* also refers to the external ORNL website used to track code development activities (see definition of *ticket*). See `https://about.gitlab.com` for more information.

ticket: a code development activity that is tracked through the stages of development, including planning, development, and review. After tickets are closed, they are archived indefinitely. Tickets are tracked on the external ORNL Gitlab website.

merge request: a request to merge new code changes into the baseline version of the code. A developer submits a merge request to an independent reviewer who must approve changes before they can be merged. Additionally, the new changeset must pass all automated tests prior to the merge.

5. QUALITY MANAGEMENT

5.1 ORGANIZATION

Each MPACT code team member shall adhere to this plan. The product software manager (PSM) retains overall responsibility for ensuring adherence to this plan. High-level VERA product suite design requirements are made at the VERA leadership level and flow down to the MPACT PSMs in the form of individual code requirements. MPACT code development, review, and testing are conducted by a small group of MPACT code developers and independent reviewers as supervised by the PSM. After approval at the code level, additional reviews and tests are conducted at the VERA product suite level under processes defined in the VERA software quality assurance program (SQAP).

5.2 ROLES AND RESPONSIBILITIES

Table 1 identifies the roles and responsibilities for MPACT software development and management activities.

Table 1. Roles and responsibilities

Role	Responsibilities
VERA program technical leader	• Approve this plan. • Define high-level software design requirements for the VERA product suite and MPACT. • Determine and/or approve the software quality level and software type. • Provide concurrence on initial software requirements documentation and all subsequent software requirement changes that are considered major software changes. • Ensure that code team members are properly qualified and trained per this plan. • Identify VERA product suite software risks and the methods to mitigate those risks that are associated with VERA product suite release.
VERA SQA coordinator	• Provide concurrence on this plan. • Provide concurrence on initial software requirements documentation and all subsequent software requirement changes that are considered a major software change. • Ensure the appropriate levels of rigor for the SQA activities. • Provide oversight to ensure that identified software problem reporting and corrective actions are implemented per this plan. • Verify that actions have been completed to properly retire the software.

Table 1 – *Continued from previous page*

Role	Responsibilities
MPACT PSM	• Perform system management and administration activities as requested by the VERA Software Integrator (VSI). • Manage code release through the VERA product suite release process. • Hold responsibility for administration and execution of this plan. • Provide task-specific training and mentoring for the code team per this plan. • Ensure that code software risks are mitigated as specified in this plan. • Address credible code software failures through validation and verification (V&V) activities. • Ensure that software configuration items are managed per this plan. • Acquire materials and services per this plan. • Assign access and authorities for MPACT repositories. • Approve software requirements and design documentation generated by code developers. • Ensure proper handling of code change requests. • Define and delineate work activities for code development and testing. • Ensure that software problems are documented and dispositioned per procedure VERA-QA-010, "Problem Reporting and Corrective Action." • Manage code software retirement when needed.
Independent technical reviewer	• Review this plan and changes to this plan. • Review requirements and design documentation. • Review and approve change requests (includes new feature implementation). • Review and approve test results. • Review and approve documentation changes associated with change requests. • Perform additional review activities as assigned by the PSM.
Software developer	• Perform and document design, coding, and unit testing of software code. • Perform additional development and testing activities as assigned by the PSM.

Table 1 – *Continued from previous page*

Role	Responsibilities
Records management officer	• Ensure that the code and documentation baseline is maintained. • Ensure that identified code records are scheduled and maintained. • Ensure that documents are numbered and controlled in accordance with Standards Based Management System (SBMS) requirements. • Maintain the documents repository to ensure easy retrieval for use and future revisions.

6. DOCUMENTATION AND RECORDS

All documents and records generated through this plan are managed per VERA-QA-003, *VERA Document Control and Records*. All MPACT documentation is retained in the MPACT code repository and is updated as necessary when modifications are made to MPACT software. MPACT change request reviews include a review of documentation. The following lifecycle documentation will be managed and maintained as QA records for MPACT:

- Software Management Plan
- Software Requirements, Test Plan, and Test Report
- Validation and Verification Manual
- Programmer's Manual
- Theory Manual
- User's Manual
- Completed software quality assurance (QA) checklists
- Software problem reports

7. SOFTWARE RISK MANAGEMENT

The software risk management approach described in VERA-QA-002, *VERA Software Quality Assurance Plan*, has been reviewed. The stated risks are applicable to the MPACT product, and no additional software risks have been identified. The software work activities implemented by the code team will ensure that the software meets its intended function before its release to the nuclear industry.

7.1 SOFTWARE QUALITY LEVEL

MPACT has been determined to be software quality level 1 (SQL1) based on its planned use for nuclear reactor safety analysis. This plan defines how the following software-specific work activities will be implemented for the MPACT product:

1. Configuration Management
2. Software Acquisition
3. Software Requirements
4. Software Design
5. Software Verification and Validation
6. Software Problem Reporting and Corrective Action
7. Training

7.2 RISK MANAGEMENT

Software risk management, which includes software failure analysis, provides a disciplined environment for proactive decision making to continuously assess what can go wrong, to determine what risks are important to address, and to implement actions to address those risks. MPACT is in the maintenance phase of the development life cycle, which requires frequent changes to support the next VERA product suite software release. The PSM is responsible for managing identified code-level risks that could impact successful completion of the next VERA product suite software release:

Introduction of code defects. Mitigating factors include software design and in-process code reviews, training and qualification of code developers, and PSM mentoring and oversight.

Inadequate resources for completion of software requirements implementation. Mitigating factors include the establishment of technical and resource-loaded performance parameters and milestones that are tracked and correlated with cost and schedule data.

8. SOFTWARE CONFIGURATION MANAGEMENT

Software configuration management will be implemented in accordance with VERA-QA-004, *VERA Software Configuration and Control*. The software configuration list can be found in the MPACT software requirements, test plan, and test report document. Software configuration management includes the unique identifying versions of MPACT and its supporting libraries and software, the name of the testing machine, and the testing results.

9. SOFTWARE ACQUISITION

Acquired software—including system software, software tools, and third-party libraries—is obtained through the infrastructure team and is controlled under the ORNL SBMS Acquisition Management System.

Inclusion of acquired third-party libraries to support the functionality of a VERA software product are covered through the software lifecycle activities, including requirements definition, design reviews, testing, and verification and validation.

10. SOFTWARE ENGINEERING METHOD

The tasks delineated in the following subsections encompass software engineering activities performed throughout the lifecycle of a software product. MPACT is already developed, has an established operational history, and is being maintained, although occasional enhancements are made, and new features are added.

10.1 REVIEWS

Reviews are conducted in accordance with VERA-QA-009, *VERA Release Process*.

10.2 SOFTWARE REQUIREMENTS

Many MPACT specific software requirements originate from CASL challenge problems, progression problems, and the upper-level VERA product suite software requirements. Flow-down from VERA product suite requirements to code-level software requirements is traced through test cases to ensure completeness of overall VERA product suite capabilities.

Software requirements have unique identifiers and are captured in a set of software requirement lists stored in the MPACT repository. A list of requirements is stored for all software packages in MPACT and is updated when software changes are made that result in a change in current requirements. A MPACT test can have one or more associated requirements identifiers. See the “Software Requirements Review” section of this plan for additional information on requirement reviews.

10.2.1 Modeling Assumptions and Constraints

Code modeling assumptions and constraints are discussed in the *MPACT Theory Manual* and the *MPACT User Manual*.

10.2.2 Requirements Traceability

A custom-developed requirements traceability tool is used to link the unique requirements identifiers to tests that evaluate those requirements. This allows for backward and forward traceability.

10.3 SOFTWARE DESIGN

Software design follows VERA-QA-006, *VERA Software Requirements and Design Process*. MPACT does not have a single design document. Code design for code changes is discussed in Gitlab tickets, which must be reviewed by an independent reviewer when they are complete, and they must be reviewed and closed by the PSM or delegate after the independent review is complete. Closed Gitlab tickets are archived indefinitely. See the “Software Design Review” section of this plan for additional information on design reviews. Code models and numerical solutions are discussed in the *MPACT Theory Manual*. A high-level overview of code organization is given in the *MPACT Programmer’s Manual*.

Design traceability is maintained by adding a Gitlab ticket number to code changes and/or merge requests to describe motivation, design, and testing related to the code changes.

10.4 SOFTWARE IMPLEMENTATION AND CONTROL OF LIBRARIES

MPACT software is implemented as outlined in the “Configuration Change Control” section of this plan using the designated MPACT coding practices. Third-party code (functional libraries such as solvers) and data (nonfunctional) libraries are utilized by MPACT and are obtained from the VERA development environment. Regardless of type, all libraries are managed and controlled by the Infrastructure Team. MPACT libraries are managed in accordance with the “Software Configuration management” section of this plan. See the “Coding Standards, Practices, and Conventions” section of this plan for additional information.

10.5 SOFTWARE TESTING

Testing is planned and conducted in accordance with VERA-QA-006, *VERA Software Requirements and Design Process*, and VERA-QA-009, *VERA Release Process*.

10.6 CODE AVAILABILITY FOR VERA PRODUCT SUITE INTEGRATION

The master branch of the MPACT Gitlab site is always available for VERA product suite integration. The master branch is synced to the casl-dev repository nightly using an automated sync script. The sync script performs automated casl-dev system testing at the VERA product suite level prior to syncing the MPACT Gitlab site with the casl-dev repository. VERA product suite releases are made from the casl-dev repositories.

10.7 SECURITY, BACKUP, AND RECOVERY

Code access is controlled through the MPACT Gitlab site. Access must be explicitly granted on a per-user basis by the PSM. Code backup and recovery services are managed by ORNL Information Technology Services.

10.8 SOFTWARE RETIREMENT

MPACT is an active software product with ongoing support. This section will be updated to reflect ongoing support, and in the event of retirement, it will be removed from distribution and support.

11. VERIFICATION AND VALIDATION

Verification and validation activities are planned and executed in accordance with VERA-QA-006, *VERA Software Requirements and Design Process*, and VERA-QA-009, *VERA Release Process*.

12. CODING STANDARDS, PRACTICES, AND CONVENTIONS

Coding standards, practices, and conventions are designed to meet code requirements, to facilitate software life cycle activities, and to improve coding consistency, functionality, performance, and testability. MPACT coding standards are documented in the *MPACT Programmer's Manual*. The MPACT PSM is responsible for managing the *MPACT Programmer's Manual*.

13. SOFTWARE TOOLS AND SUPPORT SOFTWARE

Support software consists of software tools and system software. All MPACT support software is managed and controlled by the Infrastructure Team. The PSM determines the need for MPACT support software changes and communicates it to the infrastructure team, assesses the potential impact on the code, and establishes the level of review and retesting required. Changes to MPACT support software are controlled under the VERA-level configuration management provisions.

MPACT support software is managed in accordance with the “Software Configuration management” section of this plan.

14. PROBLEM REPORTING AND CORRECTIVE ACTION

Problem reporting and corrective action follow VERA-QA-002, *VERA Software Quality Assurance Plan*; VERA-QA-004, *VERA Software Configuration and Control*; and VERA-QA-010, *VERA Problem Reporting and Corrective Action*.

15. TRAINING

Personnel performing or managing activities included in this plan shall receive training commensurate with their job responsibilities. All personnel fulfilling the roles identified in the “Roles and Responsibilities” section of this plan shall be qualified and trained as directed by the PSM. New software developers are required to review the VERA QAPP and associated procedures, this software management document, and the *MPACT Style Guide*, which is contained in the MPACT Programmer’s Manual. The following user support materials and training materials will be developed and supplied as part of the MPACT product release:

- MPACT User’s Manual
- MPACT Theory Manual
- MPACT Programmer’s Manual
- MPACT Verification and Validation Manual
- MPACT Software Requirements, Test Plan and Test Report