NRMRL QAPP REQUIREMENTS FOR SOFTWARE DEVELOPMENT PROJECTS

GENERAL REQUIREMENTS: Include cover page, distribution list, approvals, and page numbers.

0. COVER PAGE

Include the Division/Branch, project title, EPA technical lead, QA category, organization responsible for QAPP preparation, and date.

1. PROJECT DESCRIPTION AND OBJECTIVES

1.1 Describe the software and its intended application.
1.2 State the purpose of the project and list specific project objective(s).

2. ORGANIZATION AND RESPONSIBILITIES

2.1 Identify all project personnel, including QA, and related responsibilities for each participating organization, as well as their relationship to other project participants.
2.2 Include a project schedule that includes key milestones.

3. FUNCTIONAL REQUIREMENTS

3.1 Provide a list of the most important functions that the software system must address.
3.2 Identify requirements for functionality, external interfaces (includes graphical user interfaces and interfaces which are needed for other programs to call subroutines from the software, as applicable), performance, and design constraints. Each requirement should be uniquely identified and defined such that its achievement is capable of being objectively verified and validated.
3.3 Specify computer hardware and operating system requirements.

4. SYSTEM DESIGN

4.1 Provide an overview of the system design (e.g., block diagrams showing relationships between major program modules, hardware devices, and data input/output).
4.2 Describe the components and subcomponents of the software design, including databases and internal interfaces. The description should link the software structure to the functional requirements.
4.3 Provide the rationale for selecting the proposed hardware and software tools.

5. IMPLEMENTATION

5.1 Describe how a working software system is developed from the design specifications.
5.2 Describe how the requirements for functionality, external interfaces, performance,

and design constraints will be verified and validated.

5.3 Describe how release and delivery of the product is managed, including versions for alpha and beta testing.

5.4 Describe the procedures for controlling, documenting, and archiving all significant changes to software and hardware.

5.5 Identify the archiving software used for controlling, documenting, saving, and recovering changes made to the source code.

6. VALIDATION, VERIFICATION, AND TESTING

6.1 Describe the testing strategy that will be used along with the procedures for each planned test. Testing may include individual module tests, integration tests, system testing, acceptance testing, and alpha and beta testing.

6.2 Describe the procedure for checking the correctness of outputs.

6.3 Describe how it will be determined if the developed software product conforms to requirements, and whether the software product fulfills the intended use and user expectations. This includes analysis, evaluation, review, inspection, assessment, and testing of the software product and the processes that produced the product.

7. DOCUMENTATION, MAINTENANCE, AND USER SUPPORT

7.1 Specify the requirements for project documentation (e.g., requirements and design document, configuration maintenance plan, operations manual, source code, user's guide).

7.2 Describe the procedures for maintenance and user support when software or data generated by the project will be distributed outside NRMRL.

7.3 Define the methods and facilities used to maintain, store, secure, and document controlled versions and related artifacts of the identified software during all phases of the software life cycle.

8. REPORTING

8.1 List and describe the deliverables expected from each project participant.

8.2 Specify the expected final product(s) that will be prepared for the project (e.g., software, user documentation, user interface).

9. REFERENCES

Provide references either in the body of the text as footnotes or in a separate section.