

NATIONAL WEATHER SERVICE INSTRUCTION 80-304

October 28, 2004

Science and Technology

Systems Engineering

SOFTWARE DEVELOPMENT

NOTICE: This publication is available at: <http://www.nws.noaa.gov/directives/>

OPR: W/OST3 Deirdre Jones

Certified by: W/OST Frank Kelly

Type of Issuance: Initial

SUMMARY OF REVISIONS: None. This is the initial issuance.

SUMMARY: The purpose of this instruction is to define standards for the deliverables required to provide high quality, sustainable software that meets the requirements. It allows for a variety of software development models since various ones are used for software development in National Weather Service (NWS) systems (waterfall or spiral development or etc.). It is intended to cover software developed during any stage of the Operations to Services Improvement Process. This instruction is in accordance with NWS Policy Directive 80-3 *Systems Engineering*.

Signed by _____ October 14, 2004

John L. Hayes Date

Director, Office of Science
and Technology

Software Development

<u>Table of Contents</u>	<u>Page</u>
1. Introduction	3
2. Purpose and Scope	3
3. Program Product Standards	3
3.1 Design Specification Documentation	4
3.2 Source Code.....	4
3.3 Test Documentation.....	4
3.4 User Documentation	4
3.5 Administration Documentation	5
4. Program Review Standards.....	5
4.1 Design Reviews	5
4.2 Code Peer Reviews.....	5
4.4 User Reviews.....	5
Appendix A - Software Development within Context of the Operations and Services Improvement Process	A-1
Appendix B - References	B-1

Software Development

1. Introduction. Software development implements capabilities based on software requirements within the framework of the target architecture. Software development operates in conjunction with related processes: scientific research and experimentation, program and technical project management, planning and system analysis, and engineering support, integration, maintenance, and operations & support. Figure 1 in Appendix A shows software development within the context of the Operations and Service Improvement Process (OSIP).

Software development involves the following activities (IEEE/EIA 12207.0-1996):

- Process implementation and change
- Planning and management
- System and software requirements analysis
- System and software architectural analysis and design
- Detailed design
- Coding and testing
- Integration
- Qualification testing
- Transition to operations

If not stipulated by initiating plans, these activities may be structured, sub-divided, and sequenced in accordance with the software development life-cycle model applicable to the scope, magnitude, and complexity of the project.

2. Purpose and Scope. This instruction provides the context and framework for software development. It specifies software development outputs, including software, documentation, and reviews.

3. Program Product Standards. This section defines standards for deliverables to support high quality, sustainable software that meets the system requirements and user needs. Software development products can be categorized as follows:

- Design Documentation
- Source Code
- Test Documentation
- User Documentation
- Administration Documentation

Each subsection below defines standards associated with each product category.

Other applicable standards (e.g., IEEE J-STD-016, MIL-STD-1521B) will be tailored and specified by each development organization for use in their software development process.

3.1 Design Specification Documentation. Provide design specification documentation that describes:

- Developed functions
- Modules
- Concept of execution

Provide documentation that achieves the following:

- Use functional flow charts and graphics to help communicate the design
- Identify the design method if a standard one is used
- Link designs to requirements
- Identify requirements that need to be changed due to development constraints
- Distinguish between application and system layers
- Document algorithms and equations
- List inputs and outputs for each module
- Identify new software modules and those changed, and show their relationship to the overall system architecture
- List all assumptions and limitations that may apply

3.2 Source Code. Document source code by using internal comments to clearly express the functional intent of the code to a person skilled in the language and general coding practices, but not necessarily familiar with the function being performed by the code.

3.3 Test Documentation. Provide test documentation and reports in accordance with the 80-305 Test & Evaluation instruction. Users for test documentation include other testers, integrators, maintainers, and operations support personnel.

3.4 User Documentation. Provide user documentation to accompany the software in areas where there is user interaction or visibility.

Provide user documentation that:

- Describes the proper functioning of the software from a user's perspective.
Explain the operation of buttons, menu items and other techniques of user input
- Indicates any possible error states which could occur during operation

Users of this documentation include instructors who provide training and those who use the software.

3.5 Administration Documentation. Provide administration documentation that:

- Describes the configuration, setup and troubleshooting required for using the software
- Describes any pre-conditions for successful operation of the software
- Describes all options available for setup and configuration
- Describes all error modes the software could encounter and recovery procedures
- Describes the runtime footprint (e.g., memory requirements, central processing unit requirements, disk storage requirements, runtime output, and error logs) that the software leaves on the system

Users of administration documentation are system administrators and maintainers.

4. Program Review Standards. This section defines standards for software development reviews. These reviews are categorized as follows:

- Design Reviews
- Code Peer Reviews
- Test Reviews
- User Reviews

4.1 Design Reviews. Conduct one or more design reviews. Provide details appropriate to the level of complexity and potential impact to the system. The design review objective is to discuss the salient features of the design. The audience for the Design Reviews may include the engineers and designers of the collaborative software applications, the architects and those responsible for the integrity of the design, the software testers, and system integrators, and software maintainers.

4.2 Code Peer Reviews. Conduct code peer reviews, sometimes called a “peer” review or a “code walk-through.” Check for common but less obvious coding errors and correct them. Share knowledge about the code with one or more people who did not create it to serve as an additional source of knowledge about the new software. Document the code review and document follow-on actions. The audience for code reviews may include software developers and the subsystem designers for the software.

4.3 Test Reviews. Conduct test reviews. Ensure testing meets test objectives and satisfies test criteria. Identify critical and non-critical failures and remediation actions. Document the test review and document follow-on actions. The audience for test reviews may include software developers, other testers, integrators, maintainers, and operations support personnel.

4.4 User Reviews. Conduct user reviews for those software enhancements that significantly affect the way the user interacts with the system. Schedule user reviews early in design process to incorporate their recommendations into the design. Involve the range of users who may

potentially interact with the system. The audiences for user reviews are the users of the system and its designers.

5. This instruction is supported by the references in Appendix B.

Appendix A – Software Development within Context of the Operations and Services Improvement Process

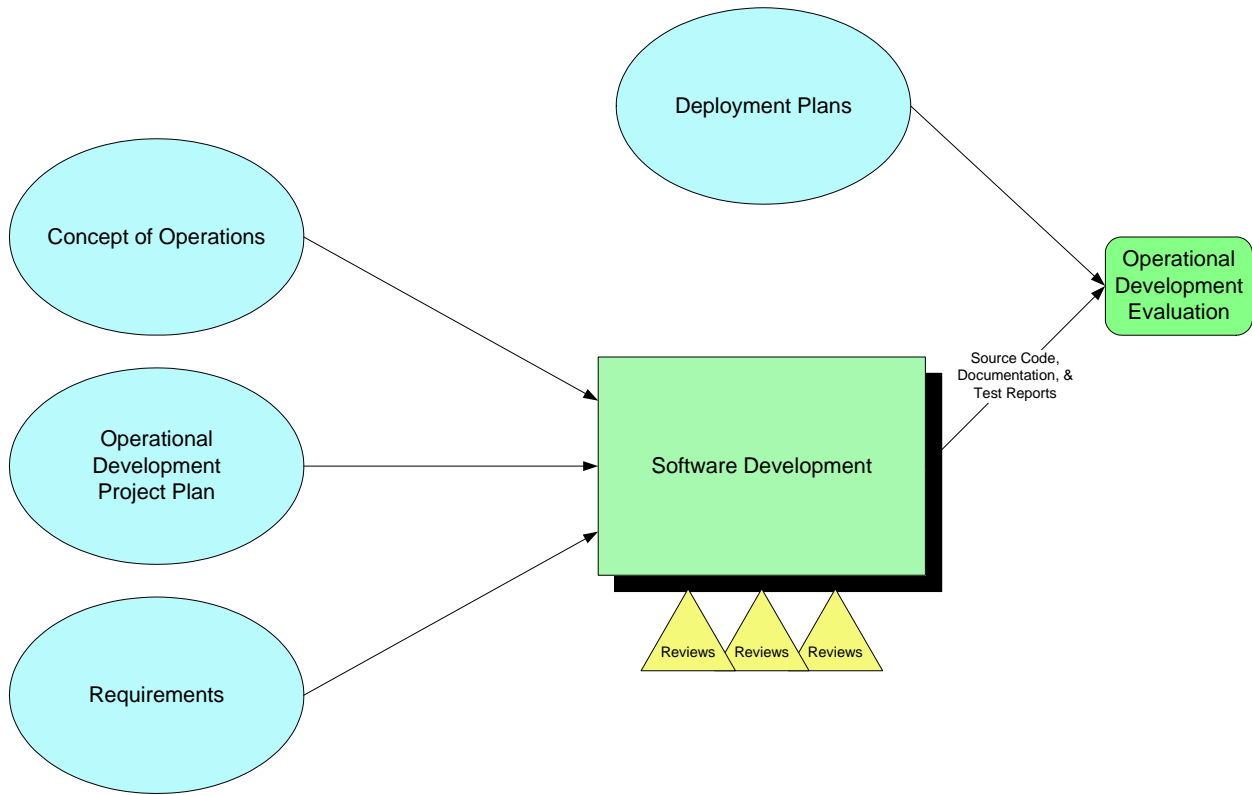


Figure 1. Software Development within Context of the Operations and Services Improvement Process

Appendix B - References

1. NWS Policy Directive 80-3, *Systems Engineering*.
2. IEEE/EIA 12207.0-1996, Industry Implementation of International Standard ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology – Software Life Cycle Processes, 1996.
3. IEEE J-STD-016-1995, *Trial-Use Standard, Standard for Information Technology Software Life Cycle Processes Software Development, Acquirer-Supplier Agreement*, 1995.
4. MIL-STD-1521B (USAF), *Military Standard Technical Reviews and Audits for Systems Equipments, and Computer Software*, 1985.