

## CODE WALKTHROUGH

### 1.0 INTRODUCTION

The Code Walkthrough is intended to ensure that the design has been correctly and completely translated into executable code and is ready for module and/or application testing. This is accomplished by a methodical examination of the software and associated documentation by developers, technical experts, and potential users. The review focuses on adherence to requirements, identification of design and coding defects, and adherence to MDL Standards, Guidelines and Procedures.

The following is a description of the Code Walkthrough methodology necessary to identify, document, manage, and resolve problems requiring change in the software developed by MDL.

### 2.0 SCHEDULE MEETING

The Code Walkthrough will be scheduled when the Task Lead and Developer determine that the code is ready for a walkthrough. The code is ready when a module or application has been compiled and built successfully.

The Task Lead or Developer will notify the Software Engineer or Branch Chief that the software is ready for a Code Walkthrough and identify suggested participants and their roles. The Branch Chief will arrange a date and place for review and formalize the participants and role assignments. The traditional roles for the code walkthrough include the Moderator, Presenter (usually the Developer), Recorder, and Reviewers. However, for smaller code walkthroughs, the Developer will serve as Moderator, Presenter and Recorder.

### 3.0 PREPARE PACKAGE

The Developer will assemble a review package that includes:

- a Code Walkthrough Meeting Form (Exhibit 1) with the name of the code, lines of code and language, the place, time and location of the meeting, date the package is distributed, and the name of each role entered;
- a compiled output with cross-reference listing from baseline code. The code listing should include individual line numbers and page numbers. The listing format should be 2 facing, numbered pages per sheet of paper in landscape mode. Use the command ***xa2ps -n filename*** to get the proper format;

- any design artifacts (requirements, site level diagram, data flow and structure charts, data dictionary listing [optional], list of APIs and objects/classes, list of error codes, assumptions, decisions); and
- a blank Code Walkthrough Checklist (Exhibit 2) and Code Walkthrough Reviewer's Comment Form (Exhibit 3).

This package should be delivered to the participants at a minimum of 1 week in advance of the review. Each Reviewer is required to examine the code using Code Walkthrough Checklist as a guide and record potential problems using the Code Walkthrough Reviewer's Comments Form.

#### **4.0 CODE WALKTHROUGH**

At each code walkthrough, the Presenter will guide the Reviewers through each section/paragraph/line of the code. The Reviewers will briefly describe the problem, and severity, type, and location of the problem indicated on their Code Walkthrough Reviewer's Comment Form. For some difficult problems, the Presenter may ask for suggestions to solve the problem. The Moderator determines how much of this discussion to allow and can terminate discussion at any time. The Recorder will document problems by assigning action items to the Developer using the Code Walkthrough Action Item Form (Exhibit 4).

Following the Code Walkthrough, the Recorder will complete the Code Walkthrough Meeting Form, Code Walkthrough Checklist, and Code Walkthrough Action Item Form and distribute this package to the walkthrough participants. The Recorder will enter action items and other appropriate information. The Recorder will also document changes to the Code Walkthrough forms (Exhibits 1-4) and submit them to the Software Manager for approval. The Moderator will ensure the package is complete and correct before distribution.

The specifics of each of the roles is described below:

- The Moderator verifies that the Reviewers have adequately prepared for the review and controls the meeting. The Moderator reminds the Reviewers of the focus; steps through each problem; calls for comments; helps solidify problems; stops attempts to resolve problems during the walkthrough; decides whether problems presented by the Reviewers are significant enough to be assigned as an action item to the Developer, Software Engineer or management; and ensures that the results are reported. If the Reviewers are not prepared, the meeting will be rescheduled.
- The Presenter is a person who is very familiar with the work and can step through each paragraph or line/section of code, presenting it at a meeting using their own words. In many cases, this will be the Developer of the code. This presentation should be done using design artifacts from the Design Review slides and Requirements Documentation (RD) in addition to the compiled output with cross-reference listing from baseline code.

- The Reviewers will usually be two to four people who identify potential software problems. These people could be technical staff members, specialists or users whose backgrounds give them insight into the material to be discussed. If supervisors fill this role, due to their expert knowledge, then it is fully understood that they are attending as peers not in the capacity of a supervisor. It may be desirable to request specific Reviewers to consider the material from different aspects, such as a user, a tester's point of view or whether the SMB Coding Standards have been applied. The potential problems are documented and classified on the Code Walkthrough Reviewer's Comments Form.
- The Recorder uses the Code Walkthrough Action Item Form to record the agreed upon action items to be assigned to the Developer. The Recorder collects the Code Walkthrough Reviewer's Comment Forms at end of meeting. The Recorder also documents any requests to change the Code Walkthrough Checklist, Reviewer's Comments Form, Meeting Form, and Action Item Form.

#### **5.0 PREPARE ACTION ITEM RESOLUTION FORM**

The assignee of the action item will document a description of the resolution for each action item using the Code Walkthrough Action Item Resolution Form (Exhibit 5). Where alternatives need to be considered, the Developer will consult the source of the assigned action item. If the correction for an action item exceeds the allocated resources for rework, or if the scope of the correction is larger than the product being reworked, the Developer will consult the Task Lead or Software Engineer (SE) and defer or omit the correction of any action item.

#### **6.0 REWORK CODE**

The Developer makes the necessary code and/or documentation modifications. Upon resolution of all the action items, the Developer will complete the Code Walkthrough Action Item Resolution Form with the date that the problem was rectified and initials the form. The Developer will distribute the completed Code Walkthrough Action Item Resolution Form to the Reviewers and other interested parties.

#### **7.0 FOLLOW-UP CODE WALKTHROUGH**

In some case, if a significant amount of rework is required, the Reviewers will have the option to request that Follow-Up Code Walkthrough be conducted on a date agreed to at the original Code Walkthrough. At this walkthrough, only the areas of the code requiring rework will be addressed.

## **8.0 RULES OF CONDUCT**

In any walkthrough, the general rules of conduct to be followed are:

- Respect each person
- Share responsibility
- Criticize items, not people
- Keep an open mind
- Question and participate
- Arrive on time
- Keep interruptions to a minimum

In addition, the following guidelines apply to a walkthrough meeting:

- Come to the meeting prepared
- Focus on the product under review
- Be concise in identifying defects
- Concentrate on the more significant issues

## **9.0 WEB SITE**

A web site exists where additional information exists concerning the code walkthrough process. The Code Walkthrough Support Page contains:

- Team Member Responsibilities
- Team Member Pool
- Code Walkthrough Documentation and Forms

The web site can be found at:

[www.mdl.nws.noaa.gov/~ifps/users\\_guide/walkthru/walk-info/codewalk.htm](http://www.mdl.nws.noaa.gov/~ifps/users_guide/walkthru/walk-info/codewalk.htm)

Exhibit 1. Code Walkthrough Meeting Form

Name of Code: \_\_\_\_\_ Date of Meeting: \_\_\_\_\_  
Library: \_\_\_\_\_ Time of Meeting: \_\_\_\_\_  
Lines of Code: \_\_\_\_\_ Location of Meeting: \_\_\_\_\_  
Language: \_\_\_\_\_ Date Distributed: \_\_\_\_\_

Role	Name	Preparation Time
Author/Developer		
Moderator		
Presenter		
Recorder		
Reviewer		
Reviewer		
Reviewer		

Number of Participants: \_\_\_\_\_ Total Preparation Time: \_\_\_\_\_

Total Meeting Time: \_\_\_\_\_

Status of Review: \_\_\_\_\_ Accept as is  
\_\_\_\_\_ Revise/No further review required  
\_\_\_\_\_ Revise/Schedule follow-up review for \_\_\_\_\_  
\_\_\_\_\_ Accept Revised Version

General Comments:

Exhibit 2. Code Walkthrough Checklist

**HM Application:** \_\_\_\_\_

**Module(s):** \_\_\_\_\_

**Requirements**

- Are all elements of the code traceable to the allocated VSRB or Design Review requirements? \_\_\_\_\_

**Code**

- Are there defects in the code? \_\_\_\_\_
- Are the modules loosely coupled? \_\_\_\_\_
- Are the modules highly cohesive? \_\_\_\_\_
- Are all error and boundary conditions properly handled? \_\_\_\_\_
- Is the code efficiently written? \_\_\_\_\_
- Does the code adhere to SMB coding standards and guidelines? \_\_\_\_\_

**In-line Documentation**

- Are the Prologues prepared according to SMB Standards? \_\_\_\_\_
- Are the Prologues accurate? \_\_\_\_\_
- Do you understand the code's intended purpose? \_\_\_\_\_
- Is the code adequately commented? \_\_\_\_\_
- Is the code accurately commented? \_\_\_\_\_

**External Documentation**

- Does the documentation accurately depict the code? \_\_\_\_\_
  - Design diagram
  - DFDs
  - API listing
  - Man Pages
  - Error Codes
  - Assumptions/Design Decisions

Exhibit 3. Code Walkthrough Reviewer's Comments Form

Name of Code: \_\_\_\_\_ Library: \_\_\_\_\_ Date of Meeting: \_\_\_\_\_  
 Reviewer: \_\_\_\_\_ Preparation Time: \_\_\_\_\_

Problem Number	Routine	Location (line #)	Description of Problem	Severity	Type

**Severity:** Severe (S), Minor (M), Cosmetic (C)  
**Type:** Design (D), Requirements (R), Logic (L), Standards (S), Data (A), Return Codes/Messages (M), Interface (I), Prologue/Comment(P)



