DO-178B and McCabe IQ

Table of Contents

Executive Summary

This document briefly describes DO-178B and how McCabe Software's McCabe IQ can be used to support the guidelines. It describes the focus of DO-178B, the Tool Qualification process in both general cases and as it relates to McCabe IQ, and the Certification Process. This document also provides a summary of McCabe IQ functionality, including specific notes about how McCabe IQ can be used to support the guidelines. Several appendices compile relevant notes to provide more information to those who are interested in this process.

This document can assist readers with becoming more familiar with DO178B, and what may be involved in qualifying McCabe IQ for airborne systems projects.

This document has been written based on:

> RTCA/DO-178B: SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION
>
>
> Prepared by: RTCA SC- 167 / EUROCAE WG-RTCA, Inc.
> Date published: December 1, 1992

Part 1: Introduction

Software Quality Assurance

The goal of the Software Quality Assurance Process is to produce high quality, risk-free software. The major tenet of the process is that to produce quality software, quality considerations must be integrated with all phases of the lifecycle process.

McCabe Software was one of the pioneers of the software quality assurance movement in the 1970s. Thomas McCabe, the founder, authored the Cyclomatic Complexity metric, widely used as the defacto standard in the industry. McCabe IQ, our flagship software QA and testing tool, provides industry leading metrics and test analysis featuring MCDC (Boolean) and path coverage technologies, among others.

Process Standards are the means by which it achieves its goals. Process standards define processes and activities necessary to produce quality software. They standardize the management and control of the software lifecycle processes.

The objectives of process standards are to:

- Establish and document a standardized and coherent *definition* of software development activities
- Provide a reference for common *terminology*, definition and vocabulary for software development
- Establish clear *expectations* between acquirer and supplier

The following is a list of some of the most recognized standards related to software lifecycle process:

Standards	Scope
ISO/IED 12207	Information technology - Software life cycle processes
ISO/IED PDTR 15504	Information technology - Software process assessment
RTCA/DO-178B	Software considerations in airborne systems and equipment certification
MIL-STD-498	Software development and documentation
ESA PSS-05	European Space Agency - Software engineering standards

Another popular framework for producing quality software is the Capability Maturity Model for Software (CMM) developed by the Software Engineering Institute (SEI). It defines a process maturity framework of five maturity levels; these levels describe an evolutionary path from ad hoc chaotic processes to mature and disciplined software processes. Although the CMM is not a standard, it has been very influential around the world for all activities related to process assessment and improvement. Detailed information about the CMM can be found at: http://www.sei.cmu.edu/

What is DO-178B?

DO-178B provides guidelines for the production of airborne systems equipment software, and for determining if an airborne software system complies with specific system airworthiness requirements. The guidelines address the concerns of the aviation industry. The aviation law does not mandate such guidelines. DO-178B:

- Is one of the most stringent standards imposed on the software development industry.
- Offers a strict certification requirement for software where anomalous behavior could cause a catastrophic failure condition.
- Is used internationally to specify the safety and airworthiness of software for avionics systems.

- Is used in the development, supply, acquisition, evaluation/certification, and operation of software products to be integrated in airborne systems and equipment.
- Describes techniques and methods appropriate to ensure the integrity and reliability of such software.
- Focuses on the entire software lifecycle environment.
- Has been used to secure FAA approval of digital computer software.
- May also be very helpful for any application where dependability is a strong requirement.
- Has been instrumental to the development of similar standards in other fields where dependability and safety is a strong requirement such as:
  Nuclear power,
  Rail and automotive transportation,
  Medical industries.
- Is published by RTCA, a private, not-for-profit organization.
- Because of its intended international audience, uses generic terms and minimizes references to specific national regulations and procedures.

Safety and Verification Focus

The DO-178B Guideline has two major areas of focus, safety and verification. Safety is the main concern of the Guideline. The Guideline defines processes with a focus on safety. The guidelines are imposed on all life cycle processes so that safety measures are designed into a product. A *System Safety Assessment* process is required to associate criticality levels with each system component.

The Guideline focuses on verification to implement safety measures in the software lifecycle processes. Particular emphasis is given to system aspects relating to software development, defining the information flow between system and software life cycle processes, mainly for safety and verification matters.

What DO-178B is NOT

It is important to be aware of what the Guideline is not about. The Guideline:

- Does not provide a complete description of the system life cycle processes, including the system safety assessment and validation processes or aircraft and engine certification processes.
- Does not cover operational aspects of software.
- Discusses certification issues only in relation to the software life cycle. This means that certification of software aspects such as user-modifiable data is beyond the scope of DO-178B.
- Does not provide guidelines concerning the structure of the applicant organizations, the relationships between the applicant and its suppliers, or how the responsibilities are divided.
- Does not provide personnel qualification criteria.

Guidelines vs. Requirements

DO-178B is a set of guidelines, NOT requirements. It has been written in general terms so that it can be used internationally. Therefore, it does not mention country specific organizations, and does not refer to country specific regulations.

DO-178B defines guidelines for software development and integral processes for defined software levels. The software levels are determined by their contribution to failure conditions as defined by the software *Safety Assessment Process*, as described below:

Software Level	Impact of failure condition on the system
A	Catastrophic
B	Hazardous, severe
C	Major
D	Minor
E	No effect on aircraft operational capability or pilot workload

Once the certification authority confirms the software as level E, no further guidelines of DO-178B would apply.

DO-178B defines guidelines for software development and integral processes. For a list of processes refer to Appendix C. The guidelines are expressed in the form of:

- *Objectives* for software life cycle processes
- Descriptions of activities and *design considerations* for achieving those objectives
- *Descriptions of the evidence* that indicates that the objectives have been satisfied

Part 2: McCabe IQ Product Suite Summary

The McCabe IQ product suite provides a broad range of capabilities for both static software analysis and test analysis. McCabe IQ provides Visualization - for quick understanding of complex software; Metrics, including the McCabe-authored cyclomatic complexity - for static and test quality information; and Depth and Breadth of Analysis - with various mechanisms and use of Structured Testing concepts.

The main products are McCabe QA, McCabe Test, and McCabe Enterprise Quality, with add-ons including McCabe EQ Style Report Designer, McCabe Slice, McCabe Compare, McCabe Data, McCabe Change, Coverage Server, and CLI Licensing Option (Automation). These features are available on multiple platforms and for multiple languages, including various legacy and today's most popular languages.

McCabe IQ Developers Edition

McCabe IQ Developers Edition is especially designed to help you assess your system's quality. You can begin your assessment by visualizing and analyzing the structure with the Battlemap, and verifying that it matches your original design. This includes both functional and object-oriented graphical representations. To assess other design aspects, you can change the criteria the Battlemap uses to assign a color to a module or class by selecting different thresholds for the metrics, or basing the module colors on different metrics.

You can use McCabe IQ Developers Edition to perform any of the following functions:

- Assess your system's design and quality using the metrics displayed in reports
- Assess quality at every level and maintain quality over time
- Obtain a composite measure of your software across the entire enterprise
- Produce industry standard report formats with a wide choice of metrics and visualization
- Generate predefined text-based and graphical quality reports, including trend reports
- Create and generate custom quality reports
- Create derived metrics and use them in reports
- Use predefined derived metrics in reports
- Save metric snapshots to perform trending of metrics over time
- If you are licensed to use the Command Line Interface, you can automate quality analysis processes

The Developers Edition includes the McCabe QA feature, and generally, depending on licensing purchased, a set of add-ons including McCabe Compare, McCabe Data, McCabe Change, and possible CLI Licensing Option (Automation).

McCabe Compare

As programs become more complex, the amount of redundant code often increases as well. In many programs, up to 40% of the code is redundant. This makes the code difficult to maintain because there is more code to manage and test; and modules with similar functionality often contain similar errors. Therefore, if you correct the error in one module, it most likely needs to be corrected in the other redundant modules. If you locate this redundant code and delete it or

reengineer it, in addition to decreasing your maintenance effort, you can decrease the size of the program, simplify it, make it more efficient, and reduce errors. If you do not locate redundant code, you may develop similar logic elsewhere, and therefore, create even more redundancy.

With McCabe Compare, you can use the module comparison tool to locate redundant code. McCabe Compare lets you select predefined search criteria or establish your own search criteria for finding similar modules. After you select the search criteria, select the modules you want to match, and specify which programs or repositories you want to search, the module comparison tool locates the modules that are similar to the ones you want to match based on the search criteria you selected.

McCabe Data

McCabe Data extends tracks data usage within your programs. As a result of the analysis of data declarations/usages and parameters, McCabe Data is able to produce metrics based on data.

McCabe Data allows you to specify a data set in the data dictionary when you want to locate one or more variables in your code and then analyze their association with the complexity of the modules in which they appear. For example, you can view the paths that contain this data, and view whether those paths have been tested. McCabe Data includes a host of tools and reports to help you locate, track, and analyze the testing of code containing the specified data set.

McCabe Change

McCabe Change identifies the modules in your program that have changed since the first time you parsed your code. In other words, if you modify the code in one of your programs and then reparse it, McCabe Change can indicate which modules contain code that was changed. You can also manually reset the change status for an individual module or for all the modules in a program and use McCabe Change to report on the change status since the most recent reset.

One of the axioms of quality assurance is that as code changes it is more likely to contain errors. You can perform an initial analysis of your code that might include generating metrics reports that indicate high complexity modules. With the addition of McCabe Change, you can, at any point in the development cycle, reparse your source code and determine which modules have been modified. As you plan QA resources, you can allocate more resources to the areas that are both highly complex and contain changed code.

You can also identify modules that changed and evaluate whether those modules are tested. You can then focus your testing on those changed modules and their interactions with the rest of your program.

McCabe Change works in combination with other McCabe IQ components as well. Using McCabe Change, McCabe Data, and McCabe Slice with McCabe IQ Enterprise Edition, you can utilize the Module Change Assessment report to determine whether modules containing changed code are also in a specific data set or transaction slice. Using McCabe Change, McCabe Data, and McCabe Slice with McCabe Test, you can utilize the Reengineering Data Coverage report to determine the testedness of changed modules that are also in a specific data set or transaction slice.

McCabe CLI Licensing Option (Automation)

McCabe CLI (command line interface) allows you to automate McCabe IQ in your environment.

CLI Licensing encourages broader tool usage across the entire organization by:

- automating similar analysis processes
- reducing the learning curve and allowing streamlined results for each user
- very efficiently using the # of users under license for the McCabe products

CLI Licensing facilitates consistency of processes within the products by:

- reducing risk of user error and lost time
- integrating with build processes
- efficiently using machine/network resources with ability to schedule in non-peak hours

CLI Licensing saves labor costs by automating repetitive tool usage processes by:

- creating multiple projects to obtain test coverage information within limitations of embedded environment, i.e. for selective instrumentation

The CLI Licensing Option is licensed based on the # of unique source files analyzed by McCabe IQ. By purchasing the CLI Licensing Option, you may optimize the use of all products across your organization.

McCabe IQ Test Team Edition

McCabe IQ Test Team Edition is especially designed to help you perform structured testing, also called basis path testing, which uses the McCabe-authored cyclomatic complexity metric to define the amount of testing any individual software module requires to be considered tested. Simply stated, structured testing requires that for a software module to be considered fully tested, a set of basis paths through the module must be tested. According to the principle of structured testing, a set of basis paths is a set of paths through a module's flowgraph in which each path is linearly independent. Therefore, any other paths through the flowgraph can be considered a combination of some subset of the basis paths. The number of basis paths through a module's flowgraph is equal to the module's cyclomatic complexity, its v(G). McCabe IQ Test Team Edition also supports other structural code coverage techniques, including line, branch and MC/DC (Boolean) coverage.

You can use McCabe IQ Test Team Edition to perform any of the following tasks:

- Prepare your source code for dynamic testing through a process called instrumentation
- Create test plans for both unit and integration/system testing
- Support analysis of unit and integration/system testing
- Design and implement test plans for MCDC (Boolean) testing
- Report on multiple levels of coverage, including lines of code, branch, path, MCDC (Boolean), data, class and architectural coverage
- Test programs written in object-oriented languages

- If you are licensed to use the Command Line Interface, you can automate test analysis processes

The Test Team Edition includes the features of the Developers Edition, plus the McCabe Test feature and McCabe Slice and McCabe Coverage Server features.

McCabe Slice

McCabe Slice is used to uncover your program's internal architecture. By compiling and running an instrumented version of your program, then importing the resulting trace file, you can visualize which parts of your program's code are associated with specific functionality.

The slice concept is important to reengineering from several standpoints.

- **Visualizing the software** - By combining the Battlemap with module coloring based on slice coverage, McCabe Slice makes it easy for you to see how your program's functionality maps to its structure.
- **Decomposing a complex system** - Using the Slice Operations option, you can combine slices generated by executing separate functions, then judge their cohesiveness.
- **Extracting business rules** - Using the Slice Operations option, you can pare down slices until they include only the specific code required for your business rules; a major step in many reengineering scenarios.
- **Tracing requirements** - By running the subset of your program's functions that satisfy a requirement, you can use McCabe Slice to produce outputs that show which code was executed.
- **Identifying dead code** - The Battlemap, combined with coverage information, can help you determine which modules in your software are never executed.

McCabe Coverage Server

McCabe Coverage Server streamlines the process for Web Testing. Your testing and QA work may now extend to web and distributed technologies including Java, C, C++, VB for EJB, Applets, Servlets, Active/X Controls, COM/DCOM and CORBA objects. McCabe Coverage Server provides support for distributed application testing with central collection of testing data and works within your browser's security restrictions.

McCabe IQ Enterprise Edition

The McCabe IQ Enterprise Edition provides for centralized reporting of metrics via an SQL database. A set of pre-defined report templates are provided. Users can organize the data from their projects in up to five levels, based on any structure convenient to the organization for reporting.

The Enterprise Edition includes features of the Developers Edition and Test Team Edition, along with the McCabe Enterprise Quality, plus following component:

Style Report Designer

McCabe IQ Enterprise Edition includes a variety of reports that you can use to gather metrics data for your project. However, you can also create your own reports using the Style Report Designer. Style Report Designer provides a visual design environment for report layout and data binding. Report templates created with the designer can be used directly through the Enterprise Quality client to generate reports.

Part 3: DO-178B Certification Process Overview

A piece of equipment that is installed on an aircraft has two major components - hardware and software. The Certification Authority considers the software as part of the total system or equipment installed. Therefore, the Certification Authority does not certify the software as a stand-alone product.

The following provides a high level description of the certification process. The process involves interactive communication between the Applicant and the Certification Authority. The process may also involve as much iteration as needed to obtain the certification.

Throughout the certification process, the Applicant should be prepared to provide additional information, data, and documentation when requested by the Certification Authority as evidence of compliance.

Certification Process Steps:

1. Applicant informs the Certification Authority of intention to certify certain equipment.

2. Certification Authority establishes the Certification Basis in consultation with the applicant. Certification Base defines regulations and special conditions that may apply to the certification process. If the equipment under investigation changes after the establishment of the Certification Base, the Certification Authority considers the impact of change and may or may not change the original Certification Base.

3. Applicant proposes a Means of Compliance that defines how the development of the equipment under investigation will satisfy the Certification Basis. The means of compliance will be embodied in the Plan for Software Aspects of Certification.

4. Applicant submits the Plan for Software Aspects of Certification to the Certification Authority. This plan includes a description of the software and the proposed software level, etc.

5. Certification Authority reviews the Plan for completeness and consistency. Reviews whether the proposed software level is consistent with system safety and data. The Certification Authority informs the applicant of the issues with the proposed plan.

6. Applicant addresses and resolves the issues raised by the Certification Authority regarding the Plan.

7. Applicant obtains agreement with the Certification Authority on the Plan for Software Aspects of Certification.

8. Applicant submits other documents to the Certification Authority. These documents provide evidence that lifecycle processes satisfy the Software Plan. The first two are the minimum documentation that should be submitted to the Certification Authority.

   Software Configuration Index
   Software Accomplishment Summary
   Executable Object Code
   Software Source Code
   Software Design Description
   Software Requirements Data

9. Applicant arranges for reviews of the software lifecycle activities with the Certification Authority.

10. Certification Authority reviews the Software Accomplishment Summary.

11. Certification Authority determines that the equipment complies with the Certification Basis.
12. Certification is granted to the Applicant for the equipment under investigation.

Part 4: DO-178B Tool Qualification Process

Some Tools Need to be Qualified

Tools can substitute or automate certain processes that are typically performed by people throughout the lifecycle process activities. When such substitution is planned, the Certification Process needs to ensure that the tool provides confidence at least equivalent to that of the processes automated. To this end, the Tool Operational Requirements document should be provided and the tool must be verified against its operational requirements. The results and the verification process should also be documented. All tool verification documents should be submitted to the Certification Authority for review and approval.

If certain features of the tool are used to substitute certain processes, only those features need to be qualified. There is no need to qualify all of the features if only a few of those features are used. Off-the-shelf software can also be used; it has to be qualified like any other tool planned to automate certain processes of the airborne system.

Project Managers are Responsible for Qualifying a Tool

Project managers and administrators (and their prime contractor) are responsible for qualifying a tool for use within their project. The applicant must satisfy certification authorities that the tool can satisfactorily perform its function within the project's development environment. The Tool Qualification Process requires the co-operation between the applicant and the tool vendor. However, tool vendors do not carry out the tool qualification process by themselves.

The applicant generates a 'Tool Qualification Plan' describing how the tool will be used by the project, what functionality is expected of the tool, and how the tool will be qualified for use with certification authorities.

The applicant indeed performs an audit of the tool to verify the tool vendors' claims. Tool vendors, in turn should prepare for the applicant's audit to show that:

- Tool has the functionality that the applicant needs
- Requirements of the tool are properly documented
- Tool, components, and documentation are controlled under configuration management
- Tool has been satisfactorily tested against its own requirements

As the four bullets above show, the audit requirements are not difficult to meet. What matters the most is the functionality of the tool for a specific project. McCabe Software can assist an applicant with such audit requirements, given the desired functionality required by the applicant.

Tools Can Be Qualified Within the Context of a Specific Project

DO-178B requires tools to be qualified for use on a particular project, within the context of that project's requirements and its development practices. A tool, therefore, cannot be qualified without regard to the project in which the tool is being used.

The rationale for the individual project-specific tool qualification is:

- Development practices vary drastically from a project to another
- A project may use a tool in a different way than other projects
- Required level of integrity is widely different among different systems

- Procedures controlling the use of the tool are as important as the tool itself. Tools used ineffectively will contribute nothing to the quality of the software. Therefore, the qualification of a tool depends on the procedures controlling the use of the tool within the project.

Therefore, when tool vendors claim that their tool is “DO-178B Qualified”, they may really mean that their tool has been qualified for a specific project.

Qualification Process Depends On the Processes That a Tool Substitutes or Automates

The Guidelines acknowledges two categories of tools: Development tools and Verification tools. Each category of tools has its own qualification process. The tools should also comply with the software configuration management and quality assurance processes of the equipment.

Because McCabe IQ is an analysis tool, it will be considered under the Verification Tools category of the Guidelines. The qualification data for McCabe IQ, and any other Verification Tool, is considered under the Control Category 2 (CC2) within the software configuration management controls. It implies such data can be placed under less stringent controls (compared to CC1) without a reduction in system safety.

Tool Qualification Process Requires Documentation

To qualify McCabe IQ, first the Plan for Software Aspects of Certification should mention that McCabe IQ is going to be qualified for specific processes, and should specify which processes McCabe IQ is going to replace or automate. The Tool Operational Requirements document for McCabe IQ should be obtained and it should be demonstrated that McCabe IQ complies with its Tool Operational Requirements under normal operational conditions.

To qualify a tool, relevant tool documentation should be prepared and included in the appropriate documentation of the airborne system.

Relevant document for the tool to be qualified	**Describes**	**Relevant document of the Airborne System**
Tool Qualification Plan	1. Data configuration category (i.e. CC2) 2. What processes it replaces 3. Tool software level 4. Tool architecture 5. Qualification activities	Plan for Software Aspects of Certification
Tool Operational Requirements	1. Tool’s operational functionality 2. User and installation manuals 3. Tool’s operational environment 4. Expected tool responses under abnormal conditions	Software Requirements Data
Tool Accomplishment Summary		Software Accomplishment Summary

The Certification Authority Must Approve the Tool

The Certification Authority must approve the usage of the tool in the airborne system. The Certification Authority in the USA is the Federal Aviation Administration (FAA).

Development tools are categorized as CC1 that implies that the Certification Authority subjects them to a more stringent review than the Verification tools that are categorized as CC2. This classification matters in two respects:

1. For verification tools, such as McCabe IQ, the above tool qualification data do not have to be documented in separate documents from the airborne system documentation. For example, Tool Qualification Plan can be just a section of the Plan for Software Aspects of Certification. But for development tools, such documents have to be prepared separately and submitted individually.

2. For verification tools, it is enough that the Certification Authority approves the Plan for Software Aspects of Certification. But for development tools, tool qualification data must be approved independently from the airborne system data.

Tool Qualification History is Important

While the fact of the matter is that the tool qualification is defined for a specific tool on a specific project, the reality is that the applicants and the certification authorities may take into account the qualification history of a tool for other DO-178B certified projects.

Part 5: McCabe IQ Product Qualification as a Tool

As mentioned before, a tool should be considered for qualification within a specific project. The Tool Qualification Process is initiated with the project managers or the administrators of the project (the applicant). The Certification Authority must approve of the use of a specific tool within a specific project. Tool vendors should assist the applicants to qualify their tools by demonstrating the compliance of their tool with the safety standards and processes of DO-178B.

The applicant should prepare a Plan for Software Aspect of Certification that communicates the means of compliance to the certification authorities.

The use of a tool as a component for the operations of airborne systems requires that tool be qualified under the System Verification guidelines. However, since McCabe IQ is an analysis tool and no subset of it is used as a component for the operations of airborne systems, the guidelines for system verification are not relevant to the McCabe IQ qualification process. The Software Verification guidelines would be sufficient for the Tool Qualification Process. The Software Verification Plan would define the means by which Software Verification process objectives will be satisfied.

This section briefly lays out the general framework for qualifying McCabe IQ. The explanatory notes address specific sections of the DO-178B document. Each point starts with a description of the relevant DO-178B guidelines, followed by a brief note on how McCabe IQ could assist with meeting the guidelines.

McCabe IQ Qualification Notes By DO-178B Section

4.1.e states the importance of defining development standards consistent with the safety objectives of the software.

> See 4.4.1

4.2.a mentions that an effective plan should provide direction to software developers.

> See 4.4.1

4.2.c states that methods and tools should be chosen that provide error prevention in the software development process.

> McCabe IQ provides an array of metrics to measure certain aspects of the code. For example, the Essential Complexity Metric is an effective measure in identifying parts of the code that have high likelihood of introducing errors during future code development activities (maintainability). Development managers should pay attention to such areas of their code and ensure that appropriate measures are taken to reduce the likelihood of introducing an error into code modules with high Essential Complexity Metric.

4.2.h concerns with the processes for deactivated code management.

> See 6.4.4.3

4.4.1 emphasizes the importance of the Development Environment in the production of high quality software.

> McCabe IQ can be used as an aid in the development process. Developers could use the tool to periodically monitor the impact of their development and enhancement efforts on the complexity of their code. If certain modifications in the code result in an unacceptable increase in the code complexity, developers can revisit their modifications and take appropriate measures. McCabe IQ

can be used as a tool for verifying the development compliance with the code complexity standards.

McCabe IQ also can be integrated with certain Integrated Development Environments, such as Microsoft Visual Studio and Eclipse. If desired, McCabe Software could develop similar integrations with other IDEs in a joint development effort with customers.

McCabe IQ provides a graphical representation of code segments that might be considered as "dead code". These are code segments that inadvertently do not have linkage to other parts of the code within an application. They would show up graphically as a stand-alone piece of code without any calling relationship with other parts of the code. Dead code often exists in large legacy applications. They exist without any purpose in the application while no one is aware of their existence. Development managers can use McCabe IQ to help identify and verify such pieces and remove them from the application.

4.4.2 states that the software verification activities need to consider particular features of the programming language and compiler.

McCabe IQ can be applied to a large number of major languages (e.g., C, C++, Ada, Java, etc.) and considers numerous dialects (implemented by different compilers) within each language.

4.4.3 states that the Test Environment Plan should define the methods, tools, procedures and hardware that will be used to test the outputs of the integration process.

McCabe IQ can provide software integration test plan information. It identifies "sub-trees" that define the integration paths, and provides the conditions in each module that enforce the execution of each sub-tree. It provides the integration test plan information in both text and graphical format.

McCabe IQ analysis can be performed on customer-defined test environments such as the development environment, target environment, emulated target environment, or host computer simulator. See also 6.4.1.

4.5 relates to the necessity of software development standards for Requirements, Design, and Coding and how they should be verified under the Verification Process.

See 4.4.1

4.6 reiterates the importance of reviews to ensure that plans and standards comply with the guidelines, to provide means to execute them.

See 6.3.4

6.3.4 deals with the reviews and analyses of the source code to detect and report errors introduced during the software coding process.

While McCabe IQ is not a debugger, it facilitates the code review and code analysis process through its graphical and visualization capabilities of the source code and a mapping of graphical representation of code structures to source lines of code.

The Battlemap graphically represents the functional hierarchy of the application, visualizing the depth of the function calls, the degree of "fan-in" and "fan-out" for each module, and the relationships among the modules within the application. The Battlemap has a similar capability for visualizing class relationships for object oriented applications.

The Flowgraphs graphically represent the control structure within each function. They facilitate code reviews and the verification of the compliance of coding practices with certain standards and guidelines.

The Annotated Source Listing in conjunction with the Flowgraphs provides a mapping between the actual source line of code and its graphical representation. Code reviewers can use the Flowgraphs to identify complex code constructs and view the actual line of code through mapping.

McCabe IQ Data capabilities provide the visibility into the data variables used in the source code. For example, it identifies the extent of the use of global variables and identifies the type of local variables and highlights specified data variables graphically for a visual analysis of the scope of a change of a data structure.

McCabe IQ graphical and textual reports also facilitate the process of code review and the code analysis process.

6.3.6 "Review and Analysis of the Output of the Integration Process" involves reviews and analysis of test cases, test procedures, and test results.

McCabe IQ generates a test plan that includes unit test cases (white box testing) and integration "sub-trees". The test cases provide the conditions to be satisfied so that certain paths within the source code are executed (tested). Test cases (test paths) can be graphically viewed. If testers use the McCabe IQ provided test cases and fully execute them, they can be assured of close to 100% testedness of their application. (Note: the source code may contain some unreachable paths that prevent the achievement of 100% testedness.) The test cases can be provided for MC/DC or for basis path testing.

Upon execution of test cases, McCabe IQ automatically documents the test results in a text file that will be imported into the McCabe IQ analysis environment for the analysis of the test coverage. This analysis can be done both graphically and through text reports. McCabe IQ displays how much test coverage has been achieved through the execution of each test plan, and identifies the areas of the code that have not been covered with any of the test plan.

The analysis of test coverage can be performed either all in one step or in a step-wise fashion where the test results are one after the other imported into McCabe IQ for a cumulative coverage analysis.

6.4 "Software Testing Process" provides guidelines for software testing at three different levels of hardware/software integration, software integration, and low level testing. It also focuses on Software Requirements Coverage Analysis, and Software Structure Coverage Analysis.

See 6.4.1

6.4.1 "Test Environment" mentions test environments that are acceptable for the Certification process.

McCabe IQ can be used to analyze test coverage achieved on any or all of the following test environments: target computer, target computer emulator, host computer simulator, development environment, or any test environment established by the Software Test Plan.

For performing the analysis based on the testing data from a target embedded system, the McCabe IQ Instrumentation Library can be customized to accommodate the memory and space limitations of the target environment. This library can also be modified to accommodate the target embedded system environment for collecting the testing results data.

6.4.2.1 "Normal Range Test Cases" deals with the requirements-based test case selection to ensure that test cases effectively test the ability of the software to respond to normal input and conditions.

Since testing all combinations of conditions is often practically impossible, McCabe IQ generates Modified Condition/Decision Coverage (MC/DC) test plan that provides Boolean test cases to verify variable usage and the Boolean operators. The MC/DC method is the most stringent testing technique available through McCabe IQ. The McCabe IQ user manuals have documented the Boolean testing methodology in detail. Line, branch, and cyclomatic basis path structural code coverage are also supported.

6.4.2.2 "Robustness Test Cases" deals with the requirements-based test case selection to ensure that test cases effectively test the ability of the software to respond to abnormal input and conditions.

While McCabe IQ is not a "test data" generator, it provides conditions for test cases that cover a linearly independent subset of paths within a module (basis path coverage) or the necessary set of MC/DC test cases for Boolean coverage. The McCabe IQ test plan ensures that each loop is executed at least once inside and once outside of the loop boundaries.

System testers should provide testing data for abnormal operating conditions. McCabe IQ would be able to import the results of such tests and demonstrate the coverage that they provide and code segments that they executed.

6.4.3 "Requirements-Based Testing Methods" elaborates on three levels of such tests and enumerates some of the typical errors to test for.

Regardless of the chosen testing methodology, as long as logic is incorporated in the source code to handle different operational conditions, McCabe IQ can analyze the results of any testing methodology. McCabe IQ static and dynamic analyses are based on the source code and testing data, compatible with the tree level testing methodology mentioned here.

6.4.4 "Test Coverage Analysis" deals with two types of test coverage, Requirements-Based and Structural, required for proper coverage analysis.

McCabe IQ provides full functionality to initiate and analyze the structural coverage. It also can be used to evaluate the structural coverage of Requirements-Based test plans.

6.4.4.1 "Requirements-Based Test Coverage Analysis" deals with the objectives of such test coverage.

Testers can import the results of their Requirements-Based testing (black box testing) into McCabe IQ and analyze the scope of their coverage. McCabe IQ does not provide Requirements-Based test cases; black box test cases should be generated from an analysis of the Requirements Specification. But the structural coverage that such test cases provide can be analyzed and graphically visualized with McCabe IQ, which highlights areas of the code that the Requirements-Based test cases did not touch in the code. This might indicate deficiencies in the generation of requirements-based test cases.

6.4.4.2 "Structural Coverage Analysis" deals with objectives of such test coverage.

McCabe IQ provides test plans for a Structural coverage analysis. Conditions for unit test plans (white box testing) and integration test plans are provided. The results of such tests are automatically collected in a file. Testers would import the results file and analyze the amount of coverage provided by the test plan.

McCabe IQ accommodates different test coverage criteria such as Lines of Code Coverage, Branch Coverage, Path Coverage, and Multiple Decision/Conditions Coverage (Boolean coverage).

Coverage reports provide the size and the percentage of coverage, based on the selected criteria, for each single module (function) and for the program as a whole.

The coverage analysis can be accomplished in cumulative steps. Testers can conduct multiple test executions on different instances of their target application, in parallel or in sequence, and import the results into McCabe IQ. At each step, testers can evaluate the contribution of each test plan to the overall test coverage, and graphically show the areas of the code that one test plan covers more than another.

6.4.4.3 "Structural Coverage Analysis Resolution" explains why the requirements-based test may not provide 100% structural coverage and how the discrepancy should be handled.

As the Guidelines acknowledge, a set of Requirements-Based test cases may not provide 100% structural coverage, no matter how complete the test cases are. McCabe IQ graphically identifies the areas of the source code that were not executed with a given test plan. Testers can then obtain additional Requirements-based test cases to increase the test coverage, or analyze the uncovered code for potential dead code segments.

A concern for inadvertently executing a deactivated code segment can be addressed with McCabe IQ, by graphically showing the deactivated code segments and visually monitoring the coverage results to verify that such code does not get activated as the result of different conditions in the test cases.

McCabe Software Will Assist Its Customers Throughout the Tool Qualification Process

McCabe Software can assist with preparing more specific documentation for a project considered for DO-178B Certification, to facilitate the Tool Qualification Process. The present document should only be viewed as introductory material in the first step of that process. This is similar to what some other tool vendors have branded as "qual packs".

McCabe Software can assist the applicant throughout the process, as the applicant establishes the applicability of McCabe IQ functionality to the customer needs, prepares the Tool Qualification Plan, performs tool audit, and presents their case to the Certification Authority.

Appendix A: About RTCA, Inc.

Organization

RTCA, Inc. is a private, not-for-profit corporation. The primary sources of funding are dues paid in support of the organization from members, academic associates and international associates. RTCA products (documents such as DO-178B) are available to the public at a small cost. It is located in Washington, D.C.

RTCA, Inc
1140 Connecticut Ave., NW, Suite 1020, Washington, DC 20036
Tel: 202-833-9339, Fax: 202-833-9434
Web site: www.rtca.org

Objectives

RTCA develops consensus-based recommendations regarding communications, navigation, surveillance, and air traffic management (CNS/ATM) system issues. It functions as a federal advisory committee. Its recommendations are used by the FAA as the basis for policy, program, and regulatory decisions and by the private sector as the basis for development, investment, and other business decisions.

Organized in 1935 as the Radio Technical Commission for Aeronautics, RTCA today includes over 200 government, industry, and academic organizations from the United States as well as other nations.

Members

Member organizations represent all facets of the aviation community. Current membership includes more than 150 U.S. and more than 40 international organizations:

U.S. members include:

The FAA, Department of Commerce, NASA, Major airlines, Aviation-related associations (such as Airline Pilots Association, etc.), Aviation service and equipment suppliers (including ARINC, Boeing Commercial Airplane Group, Honeywell International, MIT Lincoln Laboratory, Lockheed Martin, MITRE, Motorola, Raytheon, and TRW).

International members include:

AirServices Australia, Airways Corporation of New Zealand, NAV Canada, Icelandic Civil Aviation Administration, Civil Aviation Bureau of Japan, Chinese Aeronautical Radio Electronics Research Institute (CARERI), Electronic Industries of Japan, EUROCONTROL, INMARSAT, IATA, United Kingdom Civil Aviation Authority.

How RTCA works

Consensus is central to the process by which RTCA develops its recommendations. When aviation communications, navigation, surveillance, and air traffic management (CNS/ATM) issues surface or when related operational concepts begin to evolve, RTCA is frequently asked to form a committee to consider the topic and provide recommendations. RTCA has two vehicles for accomplishing these tasks: Task Force, and Special Committees:

1. On a fairly infrequent basis, RTCA is asked by the Administrator of the Federal Aviation Administration to develop industry consensus on a broad gauged strategic issue. The RTCA Policy Board may then authorize the formation of a *Task Force*, which takes the form of a short, intense activity designed to develop consensus recommendations. Examples of Task Force issues include
   - Global Navigation Satellite System (GNSS) Transition and Implementation Strategy,
   - Transition to Digital Communications,

- Free Flight Implementation and Certification.

2. The more common request is for RTCA to establish a *Special Committee* to recommend Minimum Operational Performance Standards (MOPS) or appropriate technical guidance documents. RTCA's Program Management Committee (PMC) discusses the topic and, based on consensus, initiates special committee action. This activity includes selecting a special committee chairman and providing terms of reference for the activity. Since RTCA functions as a Federal Advisory Committee, formation of the new special committee, as well as all special committee meetings, are announced in the Federal Register, and all special committee meetings are open to all interested parties.

   During special committee meetings, volunteers from government and industry explore the operational and technical ramifications of the selected topic and develop consensus recommendations. These recommendations are then presented to the RTCA Program Management Committee, which either approves the special committee report or directs additional special committee work. Approved recommendations are published and made available for sale to members and to the public.

   RTCA's Digest provides an update of RTCA Special Committee activities and related subjects every two months. This publication is available via subscription. The RTCA web site, www.rtca.org, also provides easy access to special committee activities.

Accomplishments

RTCA is a channel by which government and industry work in partnership to guide the operational use of aviation systems and technology in response to airspace user needs. RTCA products are developed by issue-oriented Special Committees staffed by volunteers. Special Committee meetings are publicly announced and open to participation by anyone with an interest in the topic under consideration. Through the years, RTCA has received several awards for its service to the aviation community. The organization was awarded the 1949 Collier Trophy for "…A guide plan for the development of a system of air navigation and traffic control for safe and unlimited aircraft operations under all weather conditions." Additionally, in 1994, the FAA named RTCA as the U.S. recipient of the ICAO 50th anniversary Medal of Honor. This unique recognition identified RTCA as the single most important U.S. contributor organization to the advancement and support of civil aviation since the creation of ICAO by the Chicago Convention in 1944.

Appendix B: DO-178B Related Documents

DO-178B Revision History:

1980: RTCA/DO-178
1985: RTCA/DO-178A
1992: RTCA/DO-178B
2000: RTCA/DO-248A

- **DO-178B**
  Software Considerations in Airborne Systems and Equipment Certification
  Issued 12-1-92
  Superseded DO-178A
  Errata Issued 3-26-99
  Provides guidance for determining, in a consistent manner and with an acceptable level of confidence that the software aspects of airborne systems and equipment comply with airworthiness requirements.

- **DO-248A**
  Second Annual report For Clarification of DO-178B, Software Considerations in Airborne Systems and Equipment Certification
  Issued 9-13-00
  Supercedes DO-248
  DO-248A incorporates all of DO-248 without change; it adds a significant number of FAQs and DPs. Provides clarification and resolution of inconsistencies of the guidance material contained in DO-178B.
  DO-248A includes:

Errata Material	Contains no new or additional guidance material.
Frequently Asked Questions (FAQ)	Provide concise responses to questions posed by industry to certification authorities and others who interpret it. Used for information purposes only. Contains no new or additional guidance material.
Discussion Papers (DP)	Provide clarification for certain sections of DO-178B where clarification requires more than short answers to questions. Provided for information purposes only. Contains no new or additional guidance material.
Keyword Index	
Correlation Table	Contains a table of correlation with DO-248 and DO-178B issues

Appendix C: DO-178B Processes

The Guideline highlights certain processes and activities that should be incorporated within the software lifecycle processes. The lifecycle processes are categorized into two groups:

A) Software Development Processes

The core of the software development is done through these processes. They include planning, requirements and design specification, coding and integration.

B) Software Integral Processes

Integral processes are performed to ensure correctness, control and confidence of the other processes and their output. These processes include verification, configuration management, quality assurance, certification, and maintenance.

The following is a summary of activities that should be performed during the software lifecycle processes in compliance with the Guideline.

A. Software Development Processes

Planning

The following plans should be prepared

- Software Project Plan
- Software Verification Plan
- Software Configuration Management Plan
- Software Quality Assurance Plan
- Plan for Software Aspects of Certification

The following standards should be defined and documented.

- Software Requirement Standards
- Software Design Standards
- Software Code Standards
- Software Verification Standards
- Software Documentation Standards

Requirements Specification

- Software High-Level Requirements Document

Design Specification

- Software High Level Design Document (architecture)
- Software Detailed Design Document (or low-level requirements)

Code Development

- Software Source Code

Integration

- Executable Object Code

B. Software Integral Processes

Verification

Verifications apply to all phases of the lifecycle processes, including the “verification phase”, itself.

Requirements

- Compliance of high-level software requirements with system requirements
- Trace-ability of high-level software requirements to system requirements
- Accuracy and consistency of high-level software requirements
- Compatibility of high-level software requirements with hardware specification
- Verifiability of high-level software requirements

Design

- Compliance of low-level software requirements with high-level software requirements
- Trace-ability of low-level software requirements to high-level software requirements
- Low-level software requirements accurate and consistent
- Low-level software requirements are verifiable
- Software architecture complies with high-level software requirements
- Software architecture traceable to high-level software requirements
- Software architecture accurate and consistent
- Software architecture compatible with hardware
- Software architecture is verifiable
- Software architecture conforms with software standards

Source Code

- Source code complies with low-level software requirements
- Source code is traceable to low-level software requirements
- Source code complies with software architecture
- Source code is traceable to software architecture
- Source code is accurate and consistent
- Source code is verifiable
- Source code conforms with software standards

Integration

- Ensure testing methodology of software verification plan was used
- Ensure test cases are accurately developed to meet requirements and coverage
- Ensure verification results are correct
- Object code complies with high-level software requirements
- Object code is robust with regard to high-level requirements
- Object code complies with low-level software requirements
- Object code is robust with regard to low-level requirements
- Object code is compatible with target hardware
- Object code complies with software structure (where applicable)

Verification (Verification of Verification Process)

- Test procedures are correct
- Results are correct
- Compliance with software verification plan
- Test coverage of high-level requirements
- Test coverage of low-level requirements

- Test coverage of object code
- Test coverage - Modified condition/decision
- Test coverage - Decision
- Test coverage - Statement
- Test coverage - Data coupling and control coupling

Configuration Management

- Identification of configuration items
- Establishing baselines and trace-ability
- Establishing problem reporting procedure, change control environment, change review process, and configuration status
- Establishing archive, retrieval, and release procedures
- Establishing software load control
- Establishing software life-cycle environment control

Software Quality Assurance (SQA)

- Assuring transition criteria for software requirements process
- Assuring transition criteria for software design process
- Assuring transition criteria for software code process
- Assuring transition criteria for software integration process