

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Traceability: Re-active vs. Pro-active

Tammy Reeve
PATMOS
Engineering Services, Inc.
Tammy@Patmos-eng.com
(425)427-1956

What DO-178B Say's on the subject

- ***“Traceability - The evidence of an association between items, such as between process outputs, between an output and its originating process, or between a requirement and its implementation.”***
- ***“5.5 Traceability guidance includes:***
 - a. ***Traceability between system requirements and software requirements should be provided to enable verification of the complete implementation of the system requirements and give visibility to the derived requirements.***
 - b. ***Traceability between the low-level requirements and high-level requirements should be provided to give visibility to the derived requirements and the architectural design decisions made during the software design process, and allow verification of the complete implementation of the high-level requirements.***
 - c. ***Traceability between Source Code and low-level requirements should be provided to enable verification of the absence of undocumented Source Code and verification of the complete implementation of the low-level requirements”***

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

What DO-254 Say's on the subject

Traceability - An identifiable association between hardware items or processes, such as between a requirement and the source of the requirement or between a verification method and its base requirement.” (note different than DO-178B)

“ 10.4.1 Traceability Data

Hardware traceability establishes a correlation between the requirements, detailed design, implementation, and verification data that facilitates configuration control, modification, and verification of the hardware item.

Hardware traceability data should include:

1. **A correlation between the system requirements allocated to hardware and the requirements.**
2. **A correlation between the requirements and the hardware detailed design data.**
3. **A correlation between the hardware detailed design data and the as-built hardware item or assembly.**
4. **A correlation between the requirements, including derived hardware requirements, and detailed design data and the verification procedures and results.**
5. **The results of a traceability analysis”**

What DO-254 Say's on the subject

- **“6.2.2 Verification Process Activities**
 - **3. Traceability between requirements, implementation and the verification procedure and results should be established. Traceability should be consistent with the design assurance level of the function performed by the hardware. It is not intended to require traceability to detailed components (for example: resistor, capacitor and gate) unless required for safety considerations”**

Data Section	Hardware Life Cycle Data (See Note 1)	Objectives	Submit	Level A	Level B	Level C	Level D
10.4.1	Hardware Traceability Data	6.1.1(1); 6.2.1(1,2)		CC2	CC2	CC2	CC2

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Pro-Active Traceability vs. Re-Active

- **Pro-Active traceability** begins at the start of the system level requirements definition and is integral to the development life cycle
 - It involves all the personnel responsible for the development of the product
 - It is integrated into the life cycle data and processes
- **Re-Active traceability** is performed once the product development and testing is complete
 - It involves one poor soul who gets to make a road map of the life cycle data. (usually a one-directional matrix)
 - If there are holes found, many times you are forced to revise released data, change code, add tests and start the entire checking process again to see if you hit the mark the second time around.

Pro-Active vs. Re-Active

•Is a Pro-active approach required by DO-178B or DO-254?

DO-254 section 5.1.1 Requirements Capture Objectives

*“Traceability of verification results to the hardware requirements is addressed in Section 6. **It is desirable to** establish this method of traceability during the requirement capture process.”*

DO-178B section 3.3 Transition Criteria Between Processes

*“The **transition criteria will depend on the planned sequence** of software development processes and integral processes, and may be affected by the software level. Examples of transition criteria which may be chosen are: that the software verification process reviews have been performed; the input is an identified configuration item; **and a traceability analysis has been completed for the input.**”*

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Lessons learned for any method.

- Never re-number, delete, or use the same requirement number once baseline for the associated document or process has been established.
- Transition points in the design process (baselines are established) are the points at which the traceability matrix should be updated and baselined.
- When possible, make the traceability activity part of the development life cycle and not an end of project report.

A Low-Tech solution for Traceability

Guidelines for effective traceability

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Matrix Information

- **C-Utilities which look for standard headers and key words can automate much of the work.**
 - Design Document information can be automatically derived from standard source file headers with a requirements subheading.
 - Module/file name/revision can be automatically derived from standard headers.
- **Verification traceability is a manual process usually because more than one requirement is verified by any one test or verification activity. Traceability to the verification tests or analysis should be accomplished as the verification activity is being performed in order to satisfy the requirements based testing objectives.**

The Matrix

- **One and only one**
- **Make the matrix usable at the testing level and for regression analysis by including the sentence that the “shall” is in as information in the matrix.**
- **Reflect derived requirements as additions to the matrix independent of a higher level SRD/SWRD/HWRD**
- **Safety and derived requirements can be marked with a special character “shall[1]d” for derived “shall[1]s” for safety related requirements.**

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Example: Software Matrix

Systems Requirements Document Or ICD	Software Requirements Document	Software Requirement Description	Design Description Doc	Module Name (for software)	Test Case document, revision and I.D.
5570_000 Rev A 3.1[1]	5560_000 Rev A 4.3[1]	The system better work or the boss shall[1] knock heads by:	5540_001, Rev A 5.2.14	evTHISexample, Rev -	5710_000, Rev A 4.2.7.8.T1 Rev -
5570_000 Rev B 3.2[5]	5560_001 Rev B 4.3.6[1]	The external oscillator clock shall[1] be divided down to achieve 512 samples for the selected range.	5540_000, Rev A 5.2.2	DpHOPEithishelps, Rev A	5700_000, Rev B 4.3.2.1.T2 Rev A
5570_000 Rev B 6.7[2]s ***Requirement deleted***	5560_000 Rev A 4.3.9.1[2] ***Requirement deleted***	The sum shall[2] be divided by the number of samples taken within the range dependent post filter sample clock rate.	Requirement deleted (see PR 1234)	Requirement deleted (see PR 1234)	Requirement deleted (see PR 1234)

Example Hardware PLD Matrix

Systems Requirements Document Or ICD	Hardware Requirements Document	Hardware Requirement Description	Hardware Design Description Doc	Test Case document, revision and I.D.
5560_000	5570_000	5570_000	5590_000	
5560_000 Rev A 3.1[1]	5570_000 Rev A 4.3[1]	The video Block shall[1] group multiple processes, which perform two basic functions: 1. Video Sample collection	Requirement deleted (see ECO #1234)	N/A
5560_000 Rev A 5.1[1]	5570_000 Rev B 4.3.6[1]	The external oscillator clock shall[1] be divided down to achieve 512 samples for the selected	5590_000 Rev A, 5.2.2	Software Verification Test 5710_000 Rev A 4.3.2.1T1
5560_000 Rev A 8.1[5]	5570_000 Rev A 4.3.9.1[2]	The sum shall[2] be divided by the number of samples taken within the range dependent post filter sample clock rate.	5590_000 Rev A 6.1.2	Hardware Verification Procedures and Results 5700_000 Rev A Simulation file <i>video_data_filter_output</i>

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Numbering, Insert & Delete Rules

- All requirements that are specified in the requirements document shall be unique requirements.
- Discussions of non-requirements for systems which may be included for ease of understanding, should not be tagged as requirements.
- Requirements for each HWRD/SWRD/SRD section shall begin with a new sequential number.
- Don't renumber the paragraphs of your traced documents once they are baselined.
- Any requirement that has been revised shall be marked with a change bar in the revised version or included in the revision page of the document.
- If a requirement is deleted, the requirement number shall be retained and the word "DELETED" shall be included in place of original text (for convenience all "Deleted" shall in a section paragraph can be placed at the end of that section). This will ensure that the sequential numbers are not affected and traceability is not lost.
- Inserted requirements are numbered after the previous shall number as a sub-level (i.e. 1.1, 1.2 would be inserted after shall[1] and if a requirement needed to be added after 1.1 it would be 1.1.1, 1.1.2 etc..
- Each requirement should be uniquely testable and/or traceable to design.
- Numbering Syntax: **paragraph_number.[shall_number]**
- Where the "shall" number is a sequential requirement number from paragraph section start to end, starting from one again at the next heading level.
- Requirements will be referenced by paragraph_number.[shall_number] in traceability matrix and for tests. (example 4.1.3.[1], 4.1.3.[1.1])

Example Requirements Document

3.4 Functional Characteristics

Top-level functional requirements are system features or outputs at the functional level and do not include any implementation details. The level of detail shall[1] be as appropriate to describe the functionality of the equipment to a pilot. The intent is to adequately define those system-level requirements that have a significant impact on the development effort and to avoid duplication of details that belong in the Software Requirements Document.

For example, a top-level requirement for a navigation function shall[2] state that the system will compute latitude and longitude using input from an IRS (inertial reference sensor). It shall[3] not define handling of IRS errors or response to an IRS failure unless such functions were expected to have a significant impact on the development effort.

This section shall[4] also identify performance requirements for the system, such as accuracy, response times, and qualities of a dynamic system. Performance requirements shall[5] be quantified with verifiable targets or with percentage deviation from a baseline system.

3.4.1 Reliability and Maintainability

This section shall[1] identify the target operational reliability, such as mean time between failure (MTBF) and/or mean time between unit replacement (MTBUR), and any in-service maintenance requirements for the product. Operational reliability and in-service life shall[2] be expressed in terms of flight hours.

Maintenance requirements include the desired diagnostic functions/features of the product that would aid a mechanic troubleshooting the system in the aircraft. Requirements for built-in test equipment (BITE) failure detection and false alarm rate shall[3] be specified. Any special tools required by line maintenance mechanics for maintaining the product shall[4] be identified

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

Example After Insert and Delete

3.4 Functional Characteristics

Top-level functional requirements are system features or outputs at the functional level and do not include any implementation details. The level of detail [shall\[1\]](#) be as appropriate to describe the functionality of the equipment to a pilot. This detail [shall\[1.1\]](#) be comparable to that located in industry documents such as an FAA Advisory Circular. The intent is to adequately define those system-level requirements that have a significant impact on the development effort and to avoid duplication of details that belong in the Software Requirements Document.

For example, a top-level requirement for a navigation function [shall\[2\]](#) state that the system will compute latitude and longitude using input from an IRS (inertial reference sensor). It [shall \[3\]](#) not define handling of IRS errors or response to an IRS failure unless such functions were expected to have a significant impact on the development effort.

This section [shall\[4\]](#) also identify performance requirements for the system, such as accuracy, response times, and qualities of a dynamic system. Performance requirements [shall\[5\]](#) be quantified with verifiable targets or with percentage deviation from a baseline system.

3.4.1 Reliability and Maintainability

This section [shall\[1\]](#) identify the target operational reliability, such as mean time between failure (MTBF) and/or mean time between unit replacement (MTBUR), and any in-service maintenance requirements for the product.

Maintenance requirements include the desired diagnostic functions/features of the product that would aid a mechanic troubleshooting the system in the aircraft. Requirements for built-in test equipment (BITE) failure detection and false alarm rate [shall\[3\]](#) be specified. Any special tools required by line maintenance mechanics for maintaining the product [shall\[4\]](#) be identified.

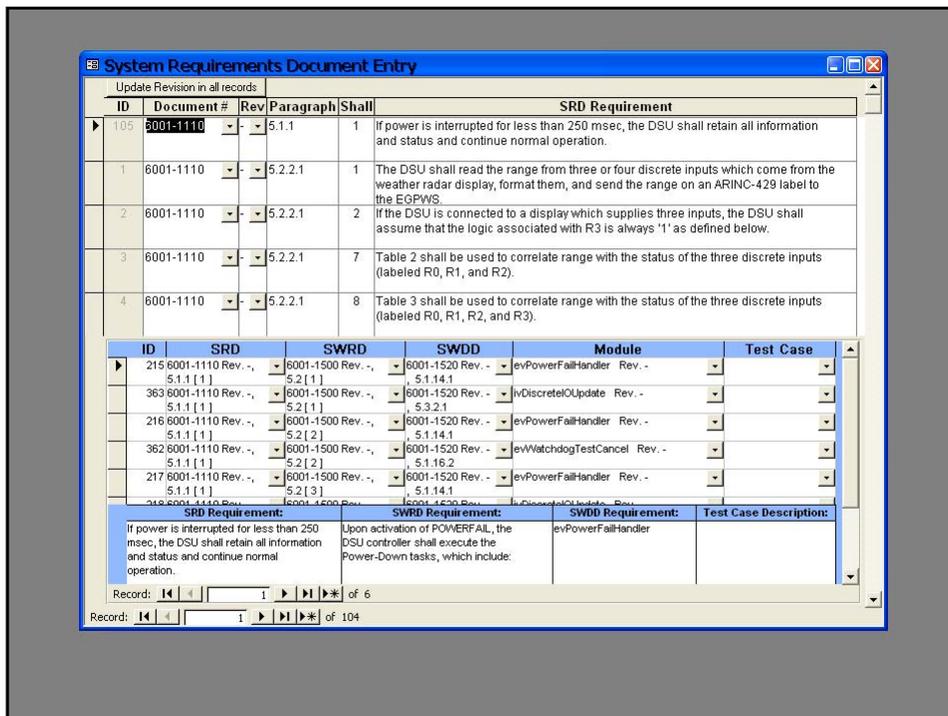
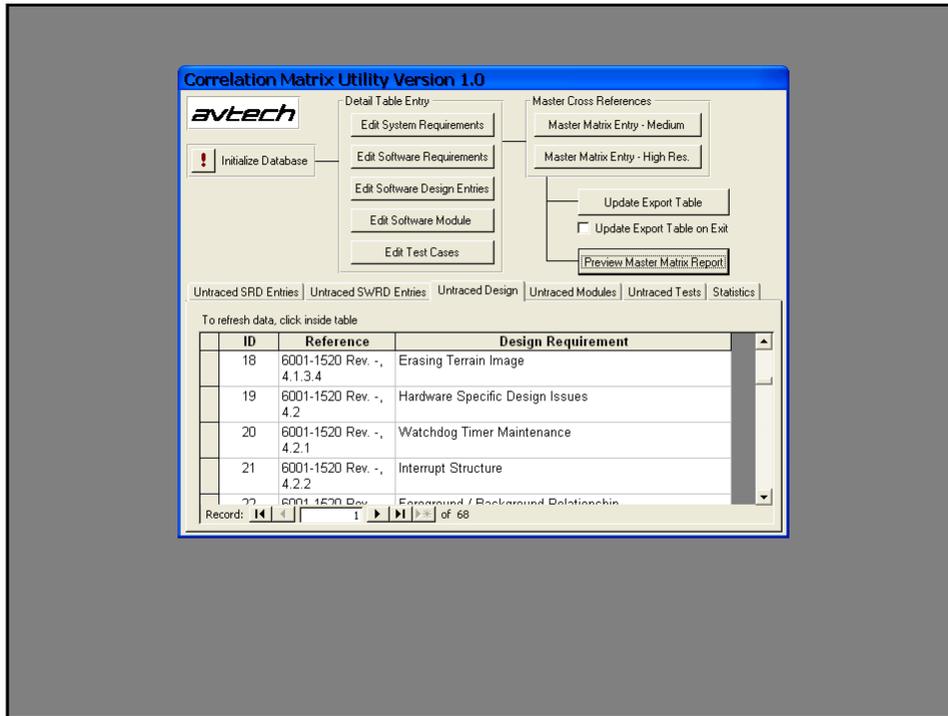
[shall\[2\]](#) deleted

Database Support

- **Access can be used to create a traceability matrix from automatically generated files.**
- *Example supplied by Avtech Corporation of Seattle WA.
Data is for example purposes only, all Rights reserved.*

2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active



2003 FAA National Software Conference

Traceability: Re-active vs. Pro-active

