

NTCIP 1203 (DMS) Deployment Experiences



T3 Webinar
September 26, 2007





Agenda

1. Version 1 Lessons Learned

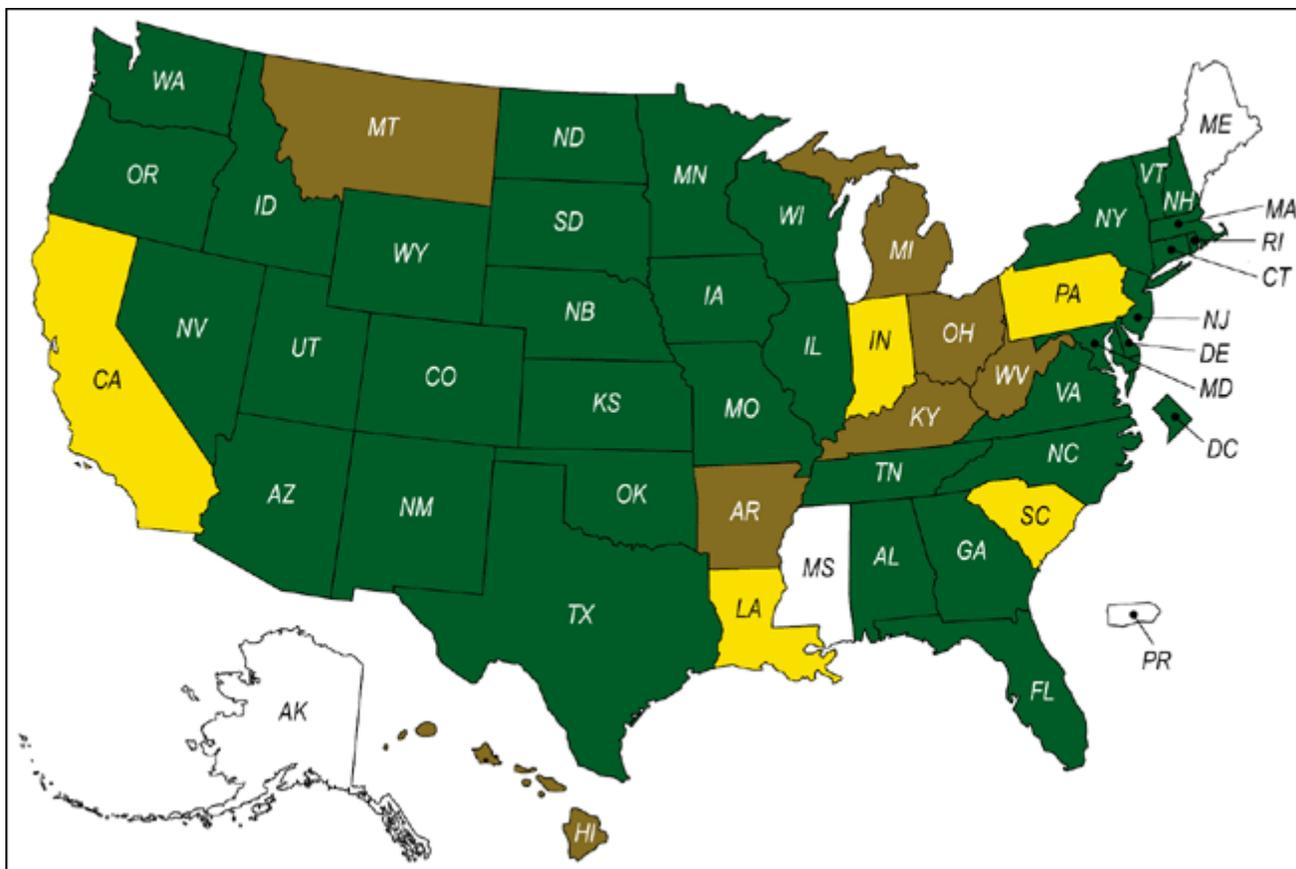
2. Version 2 Development

3. Version 2 Early Deployment



Version 1: Deployments

Version 1 Deployments (as of March 2005)



Specifications rated:

- Sufficient
- Ambiguous
- Not NTCIP
- Unknown

http://www.ops.fhwa.dot.gov/int_its_deployment/standards_imp/dms_use.htm



Version 1: Lessons Learned

- ◆ Initial deployments were painful
 - ◆ Vendors required extra time to implement
 - ◆ Implementation of standard could create bugs in software
 - ◆ Questions about interpretations of standard
 - ◆ Specifications not always rigorous enough
 - ◆ Testing, when done, required multiple rounds
 - ◆ Often accepted good-enough
- ◆ Need maturity
- ◆ Need to define a process that ensures success



Version 1: Lessons Learned

- Standard had ambiguities and omissions
 - No explicit statement of device functionality
 - But objects implied device functionality
 - Data exchange dialogs not explicitly defined
 - Can you edit a font that is in use
 - Definition of objects not always clear
 - Other omissions
- Need a way to validate and verify the standard



Version 1: Lessons Learned

- ❖ Implementers found the standard difficult to use
 - ❖ Version 1 was a design document
 - ❖ Lots of features are optional
 - 🌐 Options needed to support diversity of signs
 - ❖ Had to reverse engineer design to understand intended functionality
- ❖ Need a more user-friendly solution



Version 1: Lessons Learned

- ◆ The standard was difficult to specify
 - ◆ Had to identify each object required for project
 - ◆ Had to identify required range for each object
 - Required detailed understanding of standard
 - Made specifications difficult to understand
 - ◆ Still had to specify functional requirements
 - Conflicts between functional and NTCIP specs
 - ◆ Had to identify exact communications stack
- ◆ Need to improve quality of specifications



Version 1: Lessons Learned

- ❖ The standard was difficult to test
 - ❖ Functional requirements only implied
 - ❖ Had to derive intended processes
 - ❖ Had to define procedures
 - 🌐 ENTERPRISE/I-95 procedures became de-facto
 - ❖ Tools available to test were limited
 - 🌐 Testing required significant time
 - 🌐 Testing required extreme expertise
 - 🌐 Reproducing tests required extreme care
- ❖ Need a complete, efficient, reproducible testing solution



Version 1: Lessons Learned

- ◆ Agencies expected few problems
 - ◆ Only minimal testing was performed
 - ◆ Deployments revealed problems
 - ◆ Some problems discovered in follow-on deployments
- ◆ Agencies need to fully test each delivery



Version 1: Lessons Learned

- ❖ Deployments are not 100% interoperable
 - ❖ Deployment process is not consistent
 - ❖ Standard is not correct and complete
 - ❖ Different interpretations of standard
 - ❖ Holes in specifications
 - ❖ Inconsistent testing
- ❖ Need to create an end-to-end solution
 - ❖ Could be standardized, but not required
 - ❖ Industry needs to be aware of solution



Version 1: Deployments

- Integration is still easier
 - Standards facilitate organizational change
 - In the big picture, the problems are minor



Version 2: Development

1. Addressing Lessons Learned

2. Summary of Changes

3. Backwards Compatibility

4. Status



Version 2: Development

Lessons Learned

1. Define Process

2. V&V Standard

3. Easy-to-use

4. Improve Specs

5. Define Testing

6. Encourage Testing

7. Advertise Solution

V2 Solution

1. Follow SEP

2. Correct Standard

3. V1 Compatible

4. Develop Guides

5. Define Test Proc.

6. Testing Tools

7. Workshops & Asst

V2 Standard

ASST Tools



Version 2: Follow SEP

- ◆ Systems engineering material added to Standard
 - ◆ Concept of operations
 - User needs
 - ◆ Functional requirements
 - ◆ Dialogs
 - ◆ Detailed design
 - ◆ Traceability tables
 - Protocol Requirements List (PRL)
 - Requirements Traceability Matrix (RTM)
 - ◆ Test procedures (may be added in future)



Version 2: Follow SEP

- ◆ Extra material provides
 - ◆ Formal functional requirements
 - Removes ambiguity in previous standard
 - ◆ A more user-friendly document
 - User's select desired functionality
 - Traceability translate functions into design
 - User's need not worry about design details
- ◆ Value proven during the Early Deployment



Version 2: SEP: User Needs

- User needs define the features that may be supported
 - Activate and Display a Message
 - This feature allows an operator to activate a previously defined message to be displayed on the sign face. The message can be a blank message or come from a set of previously defined messages.
 - When activating the message the operator will need to specify the desired duration for the display and the relative priority for the proposed message to override the currently displayed message.



Version 2: SEP: PRL

- ◆ Protocol Requirements List (PRL)
 - ◆ Summarizes features defined in the standard
 - ◆ Provides a clause reference for each feature
 - ◆ Indicates whether each is optional or mandatory
 - ◆ Provides a column to select for a specific project

User Need ID	User Need	Conformance	Support
2.4.2.3.1	Activate and Display a Message	M	Yes



Version 2: SEP: PRL

Traceability to Requirements

- Many-to-many relationship
- Clause of each requirement also shown
- Conformance and Support also shown

ID	User Need	ID	Functional Requirement	Conformance	Support
2.4.2.3.1	Activate and Display a Message			M	Yes
		3.4.2.3.1	Activate a Message	M	Yes
		3.4.2.3.10.5	Retrieve a Message	M	Yes
		3.5.7	Supplemental Requirements for Locally Stored Messages	M	Yes



Version 2: SEP: Requirement

◆ Requirements define details of feature

◆ Activate a Message

● The DMS shall allow a management station to display a message on the sign face, including:

- Any permanent message supported by the sign
- Any previously defined message
- A blank message of any run-time priority
- A message based on the scheduling logic, if a scheduler is supported by the sign



Version 2: SEP: Specification

- Specifying Version 2 is primarily filling out PRL
 - Need to ensure that selections are in agreement with remainder of specification

ID	Requirement	Conform- ance	Support
3.4.1.1.1	Determine Sign Type and Technology	M	Yes
D.3.1.1	Determine Device Component Information	O	Yes / No
D.3.1.4	Determine Supported Standards	O	Yes / No



Version 2: SEP: Specification

- Some requirements require additional details

ID	Requirement	Conformance	Support	Additional Specifications
3.5.7	Supplemental Requirements for locally stored messages		Yes	
3.5.7.2	Support Changeable Messages	VMS: O.10 (1..*)	Yes / No / NA	The DMS shall support <u>32</u> changeable messages (0..65535) and <u>32K</u> bytes of changeable memory (0..4294967295).



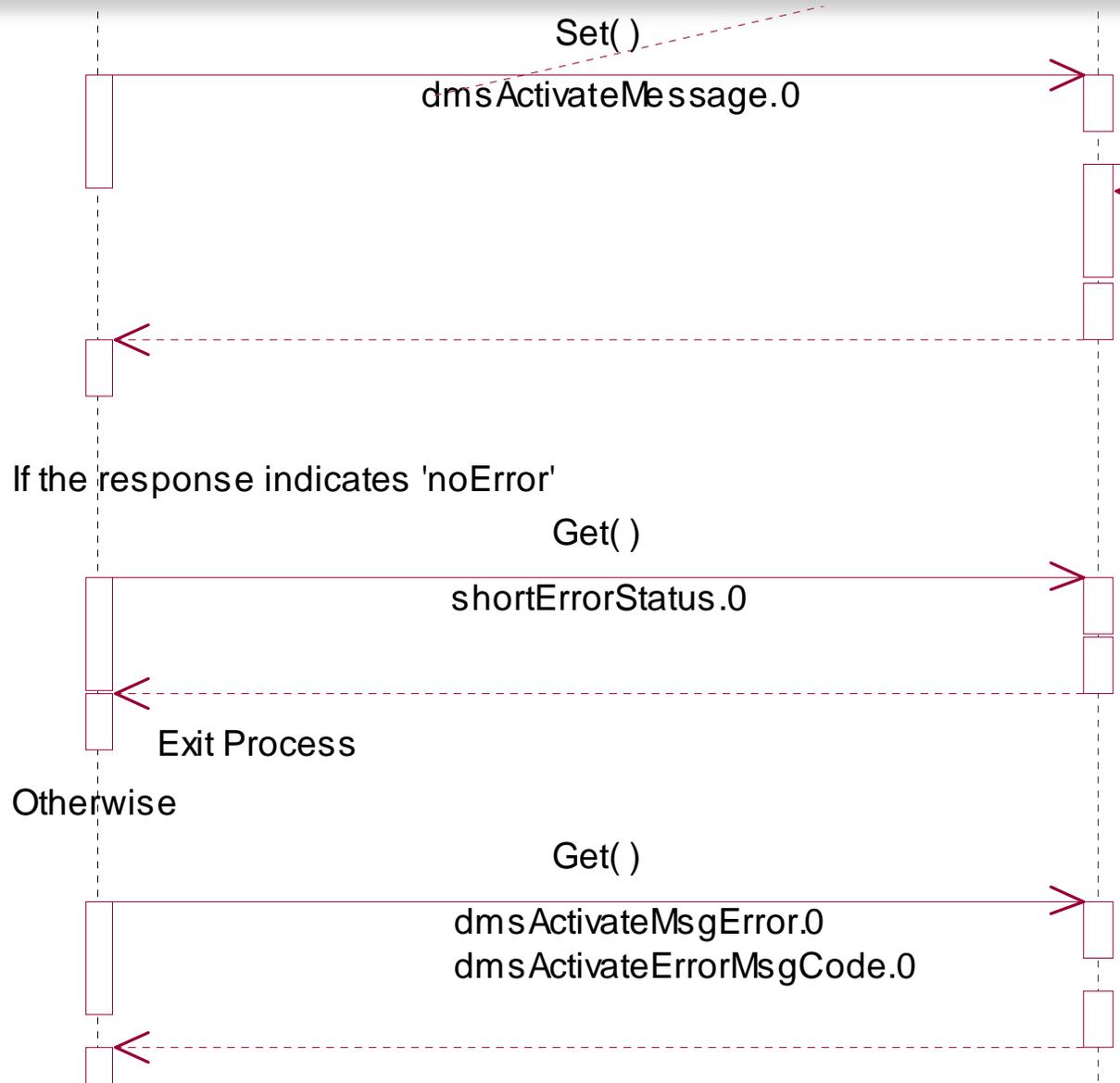
Version 2: SEP: RTM

- ◆ Requirements are traced to design details
 - ◆ Defined in Requirements Traceability Table (RTM)
 - ◆ Specification does not need to worry about RTM
 - ◆ Maps each requirement to
 - A listing of objects (essentially content of v1 standard)
 - A dialog (standardized sequence for exchanging data)

ID	Requirement	Dialog	Obj ID	Object
3.4.2.3.1	Activate a Message	4.3.2.1	5.7.3	dmsActivateMessage
			5.7.17	dmsActivateMsgError
			5.7.18	dmsMultiSyntaxError
			5.7.19	dmsMultiSyntaxErrorPosition
			5.7.20	dmsMultiOtherErrorDescription
			5.7.26	dmsActivateErrorMsgCode
			5.11.2.1.1	shortErrorStatus



Version 2: SEP: Dialog





Version 2: SEP: Object

- ◆ Objects were the only content of v1
 - ◆ Users previously had to understand this level of detail and build upwards in specifications
 - ◆ Version 1 only implied the sign functionality
 - ◆ Version 2 simplifies and tightens specifications

dmsActivateMessage OBJECT-TYPE

SYNTAX MessageActivationCode

ACCESS read-write

STATUS optional

DESCRIPTION

"<Definition> A code indicating the active message. The value of this object may be SET by a management station or modified by logic internal to the DMS (e.g., activation of the end duration message, etc.). "

::= { signControl 3 }



Version 2: SEP Summary

Benefits of SEP

- Clearly defines process used to specify product (and test...)
- Allows validation and verification of standard
- Helps resolve ambiguities
 - Ensures all dialogs are defined
 - Ensures all objects are defined
- Makes the standard more useable
 - Users can read the needs and requirements
 - Implementers can trace backwards to understand reason for objects



Version 2: Correct Standard

New Features

Graphics

24-bit Color

Msg Positioning

Critical Temp

Add'l Diagnostics

Add'l Config items

Corrections

Time

Font Definition

Brightness Ctrl

Fan Diagnostics

Changes

Auxiliary I/O

All of these support “Backwards Compatibility”



“Backwards Compatibility”

- Term applies to systems, not standards
- A standard merely “supports” the concept
 - Changes do not conflict with old mechanisms
 - V1/2 system can decode both V1 and V2 data
 - V1 mechanism is not changed
 - The heart of backwards compatibility
 - Any ambiguities still exist
 - $V1_a$ may not work with $V1_b$
- Key is to specify the desired interpretation
 - Standard can not adequately address



Version 2: V1 Compatible

	V1 _a Sign	V1 _b Sign	V1 _a /2 Sign	V1 _b /2 Sign	V2 _{only} Sign
V1 _a Central	Base Corrections* Changes*	Base	Base Corrections* Changes*	Base	Base
V1 _b Central	Base	Base Corrections* Changes*	Base	Base Corrections* Changes*	Base
V1 _a /2 Central	Base Corrections* Changes*	Base	Base Corrections Changes New	Base Corrections Changes New	Base Corrections Changes New
V1 _b /2 Central	Base	Base Corrections* Changes*	Base Corrections Changes New	Base Corrections Changes New	Base Corrections Changes New
V2 _{only} Central	Base	Base	Base Corrections Changes New	Base Corrections Changes New	Base Corrections Changes New

Assumes a common protocol stack and 1203A1 Implemented

** Features work with a manufacturer's interpretation*



Version 2: Deployment

1. Guides

2. Test Procedures

3. Test Tools

4. Workshops and Assistance



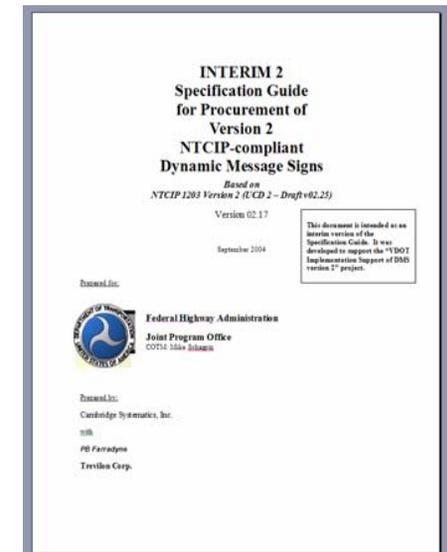
Version 2: Guides

■ Procurement Guide

- Supplements Procurement Workshop
- Explains procurement process
- Includes language to include in specification
- Includes PRL from standard

■ Workbook for Testing Workshop

- Explains testing process
- Includes sample documentation





Version 2: Test Procedures

- At least one test for every requirement
 - Tests functionality defined in standard
 - Ensures sign can display a message
 - Does not focus on accuracy of sensors
 - Does not test environmental conditions
 - Not 100% exhaustive
- Defined per NTCIP 8007 rules
 - Tool generic
 - Project generic

2.3 Test Procedures			
2.3.1 Configuration Tests			
2.3.1.1 Determine Sign Type and Technology			
Test Case:	Title:	Determine Sign Type and Technology	
1.1	Description:	This test case verifies that the DMS indicates that it is the sign type and uses the technology as required by the specification.	
	Variables:	Required_Sign_Type	PRL 2.1.2.1 and 2.1.2.3
		Required_Sign_Technology	PRL 2.1.2.2
	Pass/Fail Criteria:	The device under test (DUT) shall pass every verification step included within the Test Case in order to pass the Test Case.	
Step	Test Procedure	Device	Management Station
1	CONFIGURE Determine the enumerated value for the sign type required by the specification (PRL 2.1.2.1 and 2.1.2.3). RECORD this information as: xRequired_Sign_Type NOTE: Valid enumerated values are defined in NTCIP 1203, Clause 5.2.2 (Sign Type Parameter). NOTE: Due to an anomaly in the standard, the type field here actually references both the type and the configuration.		
2	CONFIGURE Determine the enumerated value for the sign technology required by the specification (PRL 2.1.2.2). RECORD this information as: xRequired_Sign_Technology NOTE: Valid enumerated values are defined in NTCIP 1203, Clause 5.2.9 (Sign Technology Parameter).		
3	GET the following objects: xdmSignType 0 xdmSignTechnology 0	Pass / Fail (Clause 2.4.1.1.1)	Pass / Fail (Clause 2.4.1.1.1)
4	VERIFY that the RESPONSE VALUE for dmSignType 0 is equal to Required_Sign_Type	Pass / Fail (Clause 2.1.2.1 and 2.1.2.3)	
5	VERIFY that the RESPONSE VALUE for dmSignTechnology 0 is equal to Required_Sign_Technology	Pass / Fail (Clause 2.1.2.2)	
Test Case Results			
Tested By:	Date Tested:	Pass / Fail	
Test Case Notes:			



Version 2: Test Tools

- ❖ Test procedures in formal XML structure
 - ❖ Tool-generic format
 - ❖ Allows export to automated scripts
 - Requires converter for specific script language
 - 80% automatic
 - 20% requires customization
 - Minimizes errors in implementing test procedures
 - ❖ Proof of concept included in early deployment



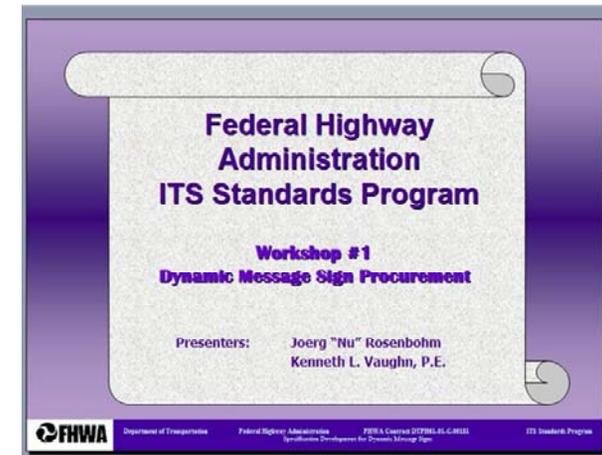
Version 2: Workshops, etc.

■ Procurement Workshop

- Explains procurement process
- Provides overview of NTCIP structure
- Explains how to specify NTCIP
- Discusses extensions to standard
- Discusses life-cycle issues

■ Testing Workshop

- Explains test documents
- Explains NTCIP details
- Explains testing process



VTTI Early Deployment





VDOT/VTTI Early Deployment

- V1 initial deployments
 - Experienced many challenges
 - Were not coordinated
- FHWA wanted a more coordinated approach
 - Demonstrate end-to-end process to industry
 - Provide feedback to standards effort
 - Provide assistance for initial deployment
 - Properly capture lessons learned



VDOT/VTTI Early Deployment

Joint effort

- Virginia DOT
- Virginia Tech Transportation Institute
- FHWA



Deployed second User Comment Draft

- One Central System
- One Sign Vendor
- Each firm was required to work in isolation
- Questions fed through Technical Assistance



VDOT/VTTI Early Deployment

Procure

RFP for Sign
RFP for Central
Evaluate Proposals
Select Vendors
Issue POs
Implement

2005-2006

Test Sign

Pre-test (Controller)
Initial Test
Final Test

Nov '06 – Feb '07

Test Central

Pre-test
Initial Test
Final Test

Dec '06 – Mar '07



User's Perspective

- ◆ PRL
 - ◆ Relatively straight forward to fill out PRL
 - ◆ Easy to make mistakes
 - Entering wrong format of information
 - Entering repeated variables inconsistently
 - PRL information drives the variable table and testing tool



VDOT/VTTI Early Deployment

- ❖ Used FHWA Test Procedures for v2
 - ❖ Tested every requirement included in the deployment (75% central/85% sign)
- ❖ Traceability tables isolated problems
 - ❖ Failures could be
 - Ambiguity in standard
 - Problem in test procedure
 - Problem in test tool
 - User error
 - Problem in device
 - Problem in central





User's Perspective

Testing

- Actual Test Case steps go above/beyond just functional tests that an agency might be used to
 - Example: Activate/display message (user need 2.4.2.3.1) has 21 steps.
 - Steps 1 and 2 are activate and display message



User's Perspective

Testing (continued)

- The RTM really does foster an amenable environment between contractors
- Eliminates finger pointing/blame game
- Applying RTM to testing the software allowed apples-to-apples comparison of the software and sign, rather than relying on strictly functional testing of the sign



VDOT/VTTI Early Deployment

- Demonstrated value of systems engineering
 - Traceability → quick identification of problems
 - Consensus because everyone can see
 - Requirement
 - Need
 - Design
 - Identification of problem → assign action item
 - Assigned action item → resolution of problems
 - Resolution of problem → accepted product
 - Accepted product avoids conflict and legal issues



VDOT/VTTI Early Deployment

Resulting tools

- DMS Procurement Guide
- DMS Procurement Workshop
- DMS Testing Workbook
- DMS Testing Workshop
- DMS Test Procedures (8007 Conformant)
- XML Version of Test Procedures
- Lessons Learned Report
- Comments back to DMS WG



VDOT/VTTI Early Deployment

- Tools still need to be updated
 - Reflect RS instead of UCD
 - Enhance based on lessons learned



VDOT/VTTI Early Deployment

Successful Deployment

Good
Standard

Good
Specs

Formal
Component
Test

Formal
Integration
Test



VDOT/VTTI Early Deployment

Ken Vaughn

Trevilon Corporation

703-390-1053

kvaughn@trevilon.com