

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Ken Leonard: ITS Standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems.

Ken Leonard: I'm Ken Leonard, the Director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS Standards Training Program. We're pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this new approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience, without the need to travel. After you complete this training, we hope that you'll tell colleagues and customers about the latest ITS standards and encourage them to take advantage of these training modules, as well as archived webinars. ITS Standards training is one of the first offerings of our updated Professional Capacity Training Program. Through the PCB Program, we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. Thank you again for participating, and we hope you find this module helpful.

Patrick Chan: Welcome to Module T311: Applying Your Test Plan to the NTCIP 12 v03 DMS Standard. What this module will cover is how to test your DMS system. The focus of this testing is going to be on NTCIP 1203 v03—how to use this particular standard for your testing—but a lot of the information in this module can apply to testing dynamic message sign systems in general.

Patrick Chan: My name is Patrick Chan. I'm a senior technical staff at ConSysTec—or Consensus Systems Technologies. I've been involved with the development of ITS Standards since 2000—specifically with the dynamic message signs—when I was still a project manager for ITS for a public agency. I've been one of the developers of NTCIP 1203 v03. I've also been involved with the development of other NTCIP standards and other ITS standards, including traffic management, data dictionaries, and connected vehicle standards.

Patrick Chan: For this module, we have four learning objectives. The first one is to describe, within the context of the testing life cycle, the role of a test plan and the testing to be undertaken for DMS. Specifically, we're going to talk about when do we test and why are we testing dynamic message signs. We're going to identify the key elements of NTCIP 1203 v03 relevant to the test plan. Specifically, we're going to identify what parts of the 1203 standard can be used to develop a test plan. We're going to describe the application of a good test plan to a DMS sign system being procured. We're going to do this by providing an example of a fictitious test plan for dynamic message signs. And finally, we're going to describe the process of adapting a test plan based on the selected user needs and requirements for an agency. We're going to do this by providing an example of how to develop test documentation based on the specific user needs and requirements of an agency.

Patrick Chan: First, we're going to talk about why do we perform testing, when do we perform testing, and what exactly are we testing when we're trying to test our dynamic message sign systems.

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: First, let's talk about why there is this module. As a procurer, operator, or specification writer of a dynamic message sign system, we need a method, a process, a plan to check that this is being provided by a vendor or the vendor fulfills all of your requirements. This module is going to walk you through elements on how to test, meaning we're going to talk about what the steps are for developing the test plan that's specific to your agency's needs and requirements. We're going to develop a test plan—a document that satisfies your agency-specific user needs, checks that all of your requirements in the specification have been fulfilled, and conforms to the appropriate standards defined in your project specification.

Patrick Chan: Why do we test? When we think about testing usually the most common answer is probably because it's a payment milestone, or that we just want to verify that something works. But technically, we perform testing to verify that the provider's system meets the procurement specs and fulfills all of the requirements in the specification. Essentially, it answers the question: Was the system that's being provided built right? Requirements were discussed and introduced in a previous module—Module A311b: Specify Requirements for DMS Based on the NTCIP 1203 Standard. To find this module, you can go the Student Supplement that was included with this module. A second reason why we test is to validate that the system satisfied the user and operational needs. Meaning—we are procuring dynamic message signs for a reason because we have a specific need. Does the system built satisfy those user needs? In other words, did I build the right system? If I have a user need to provide travel information to motorists—yes, then a dynamic message sign may satisfy those user needs. But if I wanted to be able to control a traffic signal at an intersection, for example, the system might be nice to control the traffic controller signal at an intersection, but it doesn't solve my original need which was to provide travel information to the motorist—in which case, I built the wrong system. User needs were discussed in a previous Module A311a: Understanding User Needs for a DMS Systems Based on the NTCIP 1203 Standard.

Patrick Chan: A third reason why we test is to test for conformance to the applicable standards. For this module, it's the NTCIP 1203 standard. The reason why standards are important is because we want to achieve interoperability. Interoperability is the ability of two or more systems or components to exchange information and use that information that's been exchanged. Interoperability is usually a goal of standards. An example is the Wi-Fi standard. Using the Wi-Fi standard and the other applicable Internet standards, it doesn't matter which laptop I use—whether I'm at Starbucks or the airport. It doesn't matter which router I use. Based on the Internet standards, I can get information from the Internet, or I can use it to send and get emails from the Internet, based on this Wi-Fi standard that's based on the Internet standards that we use for accessing the Internet. That's an example of interoperability. It doesn't matter who the manufacturer is of those pieces of equipment. As long as we conform to those standards, we will be able to exchange information, get our emails, get to surf the Net. The NTCIP 1203 standard is a standard that supports interoperability, but is specific to dynamic message sign systems.

Patrick Chan: This slide uses the Vee diagram to show what was taught in the ITS standards modules, and is used to demonstrate where testing belongs in the system lifecycle. The system's Vee diagram represents the entire lifecycle of a system—specifically the dynamic message sign system. This Vee diagram shows how the DMS system came about—whether it's from the regional ITS architecture or whether it's an idea that says, "We need more signs because we need to provide motorist information about the roadway." The left side of the Vee diagram represents development of the system. Start with

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

the idea and document user needs in the form of a Concept of Operations, and document in the Requirements for the system. That's followed by the development of a High-Level Design and a Detailed Design. All these concepts were already introduced in the earlier modules A311a and A311b. The next step at the bottom of the Vee is the system is then built and installed. The testing is then shown on the right side, known as the Testing Phase, and consists of Verification—Did I build the system right?—and Validation—Did I build the right system? Notice the top of the lifecycle includes Operations and Maintenance and Changes and Upgrades until the entire system is retired or replaced. For each step of the development, there's also testing that can be performed to verify or validate the system that has to be built as depicted by the lines in the middle of the Vee connecting the left side and the right side. Note that there are other modules that present more information about testing and the links to those modules can be found in the Student Supplement.

Patrick Chan: To look a little bit more into the types of testing—what are the different types of test activities that may appear in a lifecycle? The first type of testing is the Unit or Device Test Plan, where we focus on each individual component or the unit. In our case, that might be the dynamic message sign. We'll look at the dynamic message sign—specifically the LEDs, the size of the dynamic sign, the hardware, the electrical performance, and the requirements for that specific unit. Unit device testing could take place in different forms and could be called different things. For example, it could also be known as prototype testing, or design approval testing, or factory acceptance tests. The next testing activity is what we call a Subsystem Verification Plan. What we're doing here in these tests is to verify the functionality of that particular unit, but in its surrounding environment. Meaning it's not just a sign now; we want to see how to test the sign when it's connected to the communications system—whether it's a modem or whether it's a connection to your Ethernet network, when it's connected to the cabinet, and when it's connected to the permanent power supply. Subsystem verification is not only for testing the unit, but also for how it interfaces and reacts with the environment around them. Some examples of a subsystem verification test might be an onsite test. When you finally install the dynamic message sign at its permanent location, how does the test react? How does it function when connected to the modem and installed with the cabinet?

Patrick Chan: The next testing activity is what we call System Verification. It verifies the functionality of the overall dynamic message sign system. While the sign might be provided by a specific vendor, how does that sign work with the Traffic Management Center software? It might be provided by a different vendor. How do they work? Can they communicate? Can they share information? Does the system work as a whole system, as opposed to the individual parts?

Patrick Chan: The final testing activity is what we call System Validation. This confirms that the system as built satisfies the stakeholders' stated need. So does it? Did I build the right system? The system is considered validated when it's approved by the agency and the key stakeholders, whether they be the TMC operators, the users of the systems, or the maintenance people; when all of the project requirements are fulfilled—that means it has fulfilled all of the requirements in the project specification; and corrective actions have been implemented for any anomalies that have been detected. Usually, during part of a burning test—maybe it's a 90-day burning test—we find out as the operators actually operate the system and the maintainers actually maintain the system, does the system work as intended? Does it perform the functions that the operators need the system to do?

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: We talked about the testing activities. Next, we're going to talk about the Test Plan. The Test Plan is a document that documents and identifies the testing activities. It's a high-level document that identifies what is to be tested, how is the item to be tested, who is performing the testing, in what level of detail is the test item to be tested, what are the test deliverables—meaning what's going to be provided by the vendors or by the agencies—and when are the testing activities going to take place. The model that we use for developing test plans is defined in a standard called IEEE 829-2008: IEEE Standard for Software and System Test Documentation. There is also another different PCB module—Module T201: How to Write a Test Plan—that talks more, and provides more details, about test plans. The link to that module can be found in the Student Supplement.

Patrick Chan: Now we're going to go into more detail about what goes into the test plan—essentially providing more detail about the questions the test plan answers. The first is—what is being tested? The test planner identifies the scope of the test plan. Are we just testing the dynamic message sign itself? Are we testing a series of different types of dynamic message signs? Or are we testing the whole system? We not only test the dynamic message signs, but also the TMC software that's included. Note that sometimes there might be a separate test plan for each type of testing—for example, whether it's environmental testing, hardware testing, electrical testing, even structural testing—or it could be a single test plan for the entire system. That decision should be made by the agency based on the complexity and the risk involved for procuring the dynamic message signs. The higher the risk, the more likely you might want to consider a separate test plan. For example, what might be a high risk? It might be because agencies are purchasing a dynamic message sign for the first time and they have no experience with it, or perhaps no experience with the vendors, or perhaps it's a very large investment for the agency—instead of just 1 or 2 signs we're talking about 100 signs. As the risk increases, you may want to consider creating a separate test plan for each type of testing or each type of dynamic message sign. How is the item to be tested? We are identifying the test environment.

Patrick Chan: With each of the testing activities, we wanted to ask how are we going to perform that test? Is it going to be performed in a controlled environment, such as a factory or a testing lab? Or is it going to be performed after the dynamic message sign has been installed in its permanent location, assuming it's a permanent sign. NTCIP testing, which is the focus of this module, usually takes the form of interface testing, meaning that we're testing software to test the communications. We may use some kind of test software to analyze and monitor the communications between the dynamic message sign and some other system—let's say a TMC. And sometimes we may require specialized equipment to simulate environmental conditions. For temperatures testing, as an example, if you're in a laboratory, you may require a temperature chamber. In Minnesota, you might be concerned about cold weather, but in Florida or Texas, you may be concerned about hotter weather—that your dynamic message sign will operate under high temperatures.

Patrick Chan: Who is to test the items? This item identifies the roles and responsibilities for everyone that's involved with the testing. It may be a staff engineer. It might be a TMC operator. It may be the agency. It could be a consultant. And it will include the vendor. The test plan will identify the role of each of these people involved in the testing. What is it that they're supposed to prepare, what are they supposed to do, and what's their role during the testing?

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: In what detail should the items be tested? The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one. The amount of detail may be a function of the risk. For a DMS system the agency has experience with and for a vendor that they worked with before, the detail and the amount of testing might be small. For a new dynamic message sign that might have new features, or that the agency has various experience with that's high risk—either because of complexity or safety concerns—the testing activities may require additional detail. The test plan should identify the minimal degree of comprehensiveness desired. We need to identify constraints in the test plan—such as the availability of dynamic message signs, the resource availabilities, and any deadlines.

Patrick Chan: What are the test deliverables? What gets delivered at the end of the test—before the testing activity starts—and at the end of the testing activities? Who submits it? What are the requirements for each deliverable? Milestones should be included in draft deliverables to allow for a commenting period and for time to resolve issues that may appear during the testing activities.

Patrick Chan: The Test Plan is really a high-level document providing an overview of the testing activities and organizing those testing activities. The details of the actual test are found in other documents that are referenced in the Test Plan. These documents include the Test Design Specification, which provides the details of the test approach for a specific feature or combination of features. It identifies the test case specifications to be performed as part of that test. The Test Case Specification is another document that specifies the inputs, predicted results, a set of execution conditions, and a Pass/Fail criteria for the test item. And finally, the third document is a Test Procedure Specification, which specifies the sequence of actions for the execution of the test—meaning what are the steps that have to be taken to perform the test?

Patrick Chan: The reason why we have these different types of test documentation is for modularity. This slide is a graphic of the relationship between the Test Plan, the Test Design Specification, the Test Case Specification, and the Test Procedure Specification. The modularity allows the specification to be reused over and over again. For example, there may be a test specification for activating a message. The process for activating a message is defined by NTCIP 1203 and is the same, regardless of the type of sign or whether it's a new message or planting a sign or selecting a message from the library except that the inputs and expected outputs are different. Thus, there may be a test case specification and test procedure specification just for activating a message on a dynamic message sign. This test case and test procedure specification may be used over and over again by all test design specifications. There may be a test design specification for each type of test—for example, one for a hardware test or for an interface test—or there might be one for the type of sign being purchased, such as a blank out sign or a color VMS sign. Each type might have different features that might be tested or possibly have different values, but the cascade is to activate a message and the steps for activating that message are the same, regardless of the type of message. A test plan may also reference different test design specifications depending on what is being purchased at the time. For example, an agency may be interested in purchasing two types of dynamic message signs—a blank out sign and a variable message sign. There may be a test design specification for each type of sign. The test case and test procedure specification with certain functions—such as activated design—will be referenced by both, because both test design specifications will probably have activated message as part of the testing. However, there may be several test procedures and test case specifications that are unique to each type of sign, such as a VMS may have a test case for

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

supporting graphics and colors. Several years later, though, the agency may have another specification for the new set of variable message signs that are single-color signs. A new test plan will probably be developed and a modified test design specification will probably be created, but the test case and test procedures can be reused. You don't have to rewrite the test case or test procedure specification for activating a message on a dynamic message sign. There is additional information about writing these test designs, test cases, and test procedures. There are modules for each of these also. For example, there's a Module 13: Overview of Test Design Specifications, Test Cases, and Test Procedures. The link for that module can be found in the Student Supplement.

Patrick Chan: We've reached our first activity. The purpose of the activity is to revisit and reinforce what we've learned in the previous set of slides.

Patrick Chan: Our test activity is: What does a test case specification do? Your answer choices are: A) Specifies the inputs, predicted results, and the conditions for one or more functions in the test item; B) Specifies the details of the test approach for a feature or combination of features; C) Describes the scope, approach, and resources for the testing activities; or D) Specifies the sequence of actions for the execution of the test. Again, the question is, what does a test case specification do?

Patrick Chan: The correct answer for this question of what does a test case specification do is A. It specifies the inputs, predicted results, and conditions for one or more functions in the test item. B specifies the details of the test approach for the feature or a combination of features and describes the test design specification, while the test plan describes the scope, approach, and resources for the testing activities. And finally, D is actually a test procedure specification, which specifies the sequence of actions for the execution of a test.

Patrick Chan: Our next learning objective is to identify the key elements of the NTCIP 1203 v03 relevant to the test plan.

Patrick Chan: We're going to go over which parts of the NTCIP 1203 v03 standard can be used to help an agency develop a test plan.

Patrick Chan: First, let's quickly go over what NTCIP 1203 is. NTCIP 1203 is an information layer standard that specifies the interface between the Management Station or host system and a dynamic message sign in a field. It doesn't define how the data travels between the dynamic message sign and Management Station, but rather it defines the contents of the data. Think of it as a data dictionary containing the vocabulary, the words that an agency can use to monitor and control and configure a dynamic message sign.

Patrick Chan: NTCIP 1203 v01 was first published in 1999, and Amendment 1 was published in 2001. In 2010, v02 was published, and what that version did was add systems engineering content—meaning prior to v02 NTCIP 1203 was a data dictionary. It was a dictionary. It contained a bunch of data elements that could be used to monitor and control a dynamic message sign. But agencies didn't know how to use it. They didn't know which words can be used or should be used to perform a particular function. v02 added that systems engineering content. Based on an agency's specific user needs, and based on their specific requirements, the standard told an agency, and an implementer, and a vendor which data elements should be used as defined in the standard to fulfill a specific requirement. v02 also added some new functionalities—such as support for color and graphics. v03, published in 2014, took essentially v02

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

and added test cases and test procedures. There was no new functionality, but a standard size set of test cases and test procedures was added, allowing agencies to procure dynamic message signs to consistently test for conformance to the NTCIP 1203 v03 standard.

Patrick Chan: When we're talking about interface and communications testing—such as testing for NTCIP 1203—there are a couple of things we're testing. One is compliance with the procurement specification, meaning does the DMS fulfill all of the communication requirements in the agency's procurement specification. We're also testing for conformance with the NTCIP standard. In NTCIP 1203 standards v02, v03, and above, there is a tool or a table called the Protocol Requirements List that an agency can use to define the specific user needs and requirements for that agency. To conform to the standard, the dynamic message sign must fulfill the mandatory requirements as specified by the standard, in addition to any of the selected optional requirements of NTCIP 1203 in the standards that are referenced. This means if an agency says, "For my procurement, you need to support these optional requirements to conform to the standard," you must still test for those optional requirements standards. Those optional requirements are fulfilled according to the NTCIP 1203 standard. Note that conformance is not compliance. We comply with the procurement specification; we conform with the standard.

Patrick Chan: In addition, when we're performing testing for NTCIP 1203, we are testing that the communications requirements are fulfilled according to the standard. In NTCIP 1203 v02 and v03, there is another tool or matrix called the Requirements Traceability Matrix that defines how to fulfill a standard requirement—meaning to fulfill this requirement, there are specific data exchanges or dialogs. Those dialogs—those data exchanges—must occur in the sequence defined by the standard. It could be data exchanges; it could be events. The Requirements Traceability Matrix also shows that to fulfill a requirement, these are the data objects defined by the standard to fulfill that particular requirement. That's part of the communications requirement testing. When we're testing for NTCIP 1203, we're also testing that the functional requirements are fulfilled. That means is the DMS system performing the functions as expected? For example, if there's a requirement to command a DMS to display a message, does the DMS system display that message? There are also a bunch of performance requirements that can be tested in NTCIP 1203. These performance requirements indicate how well a dynamic message sign system does something—how quickly does it respond to a command, for example. And, if the dynamic message sign is supposed to support three fonts, does the dynamic message sign indeed support three different fonts?

Patrick Chan: For testing, in NTCIP 1203 v03 a third tool was added to the standard—a third matrix table. This table is called the Requirements to Test Case Traceability Matrix. What this table does is list the test cases that must be passed to fully test whether a standard requirement has been fulfilled by the implementation.

Patrick Chan: What we show here is an example of this Requirements to Test Case Traceability Matrix. We're showing that for the requirement "Activate pixel testing," which is requirement 3.5.3.1.1.2, both test cases C.3.5.1 "Pixel test with no errors" and C.3.5.2. "Pixel test with errors" must be passed to verify that the requirement is met.

Patrick Chan: Notice that for certain requirements, multiple test cases may need to be fully completed to test the requirement. Each test case may test different conditions. In the example we showed, we had a test case for does the dynamic message sign system perform properly when there is no error detected

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

and does it react correctly in a standardized way when an error is reported for a pixel test. Each test case may also test the different set of values. For example, there are requirements for left, center, right, and full justify. There might be separate test cases to verify does the left justify the text properly or does it center, or right, or full justify properly? When there are multiple test cases that are traced to a requirement, implementation must pass all of the test cases that the requirement traces to before being able to claim that the requirement is properly met.

Patrick Chan: In NTCIP 1203 v03, there are also test case specifications. Remember that the Test Case Specification is a document with specified inputs, predicted results, and the execution conditions. This information can be found in the header of each test case that's in NTCIP 1203. Note the agency may want to perform a test case specification multiple times—each iteration with a different input and with possibly a different expected output. Note that a test case specification needs to be only performed once to verify conformance to the standard. However, you may want to perform the test case specification multiple times to verify the test complies with the project specification. For example—and we'll show this in the next slide—if you have three different fonts, you may want to perform the test case three times—once for each font. Sometimes you may also want to perform a test case more than three times because you want to perform what we call “negative or exemption testing” by deliberately using invalid values to verify the dynamic message sign behavior. For example, if you want to confirm that message number two is blank, you may want to run test message number one to show this message number one came up properly and is being displayed, but there is no message number two. I want to confirm that the dynamic message sign properly rejects a command that says run message number two when there is no such thing as message number two. Testing a dynamic message sign to verify correct performance is important, but when it involves safety, you may also want to confirm that the dynamic message sign system reacts properly in case you give it a bad value.

Patrick Chan: The project specification requires dynamic message signs to be preconfigured with three fonts. To conform to the standard, I only have to perform this test case and retrieve a font definition once. But I may want to perform it three times to verify all three fonts so that I can verify that the dynamic message sign complies with my project specification that says the sign comes with three fonts.

Patrick Chan: This is an example of what the test case specification looks like in NTCIP 1203 v03. Note that even though it is v03, the same test cases, and almost all of the requirements, can also be used for the NTCIP 1203 v01 systems, and also for v02 systems. Sometime if the requirements did change and the design changed a little, it's indicated in the standard by saying this test case is only for v01 systems, or this test case example is only for v02 or above systems. Looking over the information in each test case, the first thing that we have is that there is a title. Each test case has its unique ID. The title of test case 5.1 is “Pixel test with no errors.” The test case specification includes a description of the test case. Exactly what are we testing? It also includes variables. What are the test values coming from? Are the outputs coming from another test? Is it coming from a random generator? Is there a predetermined test value? Finally, it also includes what the Pass/Fail criteria is. To pass the test case the DUT—device under test—shall pass every verification test included within the test case.

Patrick Chan: Also in NTCIP 1203 v03 are the test procedure specifications. The Test Procedure Specification is a document. It contains the sequence of actions for the execution of a test. That means it defines the steps—and only the steps—necessary to test the function. Standard test procedures ensure

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

that conformance testing is performed in the same manner on separate testing occasions. That means if I perform the same test five times, the same test procedure should be used five times. Assuming the same inputs, this results in the same exact outputs all five times. A test procedure in a test case specification may be called by another test procedure in a different test case specification. For example, we have a test procedure called blank the dynamic message sign. The test procedure is to blank the dynamic message sign so it's not showing anything. We will use this test procedure multiple times and call from another test case, another test procedure, because we want to make sure that the dynamic message sign is in a known condition at the beginning and at the end of testing. This is a blank dynamic message sign test procedure that you will use over and over again within NTCIP 1203 v03 test cases and test procedures. As a final note, it is important not to skip any steps in the test procedures to ensure compliance with conformance testing.

Patrick Chan: We have combined NTCIP 1203 v03 and all of the NTCIP standards that include test cases and test procedures into one test case. It was a decision that was made to make it easier for agencies to use the standard. The top of a test case will include information for the test case. Right after the heading information, we go straight into the test procedure. This is the heading that provides the test case specification inputs. Right below, we include the test procedure specification—or the steps to perform the specific test case.

Patrick Chan: This provides a more expanded example of what the test procedure specification may look like. On the left, we have a test step number which is a unique identifier within the test case or test procedure that defines a normal sequential order to execute a test procedures test. Skipping the Test Procedure column briefly, we have two additional columns called Results. This means after completing a test step, if it was completed properly, we can indicate yes—the test step was completed properly, in which case it would pass. Or if it wasn't completed properly, we can indicate this particular step failed. The Additional References column provides us with information in case I want to look at why we're performing this step, or find out where in the standard does it describe performing this step. The Additional References provide that information. This is the section—4.2.4.2 Step A—that indicates this particular step. Back to Pass/Fail—Section 3.5.3.1.1.2 indicates the requirement that is being met through this particular step.

Patrick Chan: The test procedure steps have several keywords that are used to indicate the steps to perform a test procedure. Some of the keywords we use include CONFIGURE. It indicates the test step is a predicate to identify a configurable variable. It's like setting up a variable and defining the value of a variable so that we can perform the test. That's the input. SET-UP indicates a test step to set up the environment for the actual test. Then finally, we also have a NOTE that provides additional information that can help you to perform the test. Note that keywords are defined in NTCIP 8007—that's included in this Student Supplement. The link to the NTCIP 8007 standard is in the Student Supplement.

Patrick Chan: Other keywords that may appear in the test procedures include VERIFY. The VERIFY keyword is part of the test that says check the value and make sure that the value is what we expected.

Patrick Chan: Another keyword that we use is PERFORM, where we perform the test case. We call a different test procedure—in this case to blank the sign. The reason why we want to do this is that when we complete the test cases, we want to make sure that we leave the dynamic message sign in a known state.

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: We've reached another activity.

Patrick Chan: The question is: What is the purpose of the Requirements to Test Case Matrix? The answer choices are: A) Identify the requirements that are part of the project specification; B) Identify all of the test cases that must be passed to verify a requirement is fulfilled; C) Identify the design content to fulfill requirement; D) Identify one of the possible test cases that must be passed to verify the requirement is fulfilled. Again, the question is: What is the purpose of the Requirements to Test Case Matrix.

Patrick Chan: The correct answer is B) Identify all the test cases that must be passed to verify a requirement is fulfilled. Identifying the requirements that are part of a project specification is the job of the Protocol Requirements List. To identify the design content that is necessary to fulfill a requirement is the purpose of the Requirements Traceability Matrix. And finally, to identify all test cases that trace the requirements must be passed to verify the requirement is fulfilled.

Patrick Chan: In our next couple of slides, we're going to describe the application of a good test plan for a DMS that is being procured. What we'll do is we'll go through an example of a Test Plan for the DMS system. Specifically, we'll demonstrate how to develop a test plan customized for an agency's specific user needs and requirements.

Patrick Chan: Using the IEEE-892-2008 specification, your test plan may include a test plan identifier—like Test Plan 2017-01 DMS. Each test plan should have a unique identifier. It should include an Introduction that describes the purpose and maybe the scope of the test plan. The purpose of our test plan is to verify compliance with procurement number 11-xxx and verify conformance with NTCIP 1203 v03. That might be the purpose of your test plan. Test Items describe what we're testing. In our example, we want to test the entire dynamic message sign system. That includes the ATMS software—you may want to indicate which version of the ATMS software we're testing. We may have two different procurements—one for blank out signs and maybe procuring five blank out signs, and this is the procurement number—and five variable message signs that might be text and contain three lines of text with 24 characters in each row. That might be procurement number X-- 11-xxx or it might be 11-yyy.

Patrick Chan: The test plan should also indicate which features are to be tested. If you've filled out a Protocol Requirements List that indicates which user needs and requirements are selected for your project specification, include it. Make that completed Protocol Requirements List part of your test plan. Your test plan shows these are my user needs that are to be satisfied, and these are my requirements that need to be fulfilled for this particular procurement. You may also want to include which features are not to be tested—but that's optional. We may also want to have a couple of paragraphs discussing your approach, how the tests are organized in your test plan, and how the results of the testing are to be logged. You may also want to include a statement about what the Pass/Fail criteria are for specific items. To pass a test, each item being tested must pass all of the test procedures associated with the requirements for the test item.

Patrick Chan: Suspension Criteria and Resumption Requirements. You want to specify a criterion to suspend all or a portion of the testing activities associated with this test plan. Invariably, when you perform testing, something unexpected is going to happen. What's the criteria for suspending the test and for restarting the test? If it's a minor thing, you may say let's restart the test from where we left off. Or if it's a major problem, you may say something happened, so we're going to restart the test from the

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

beginning—meaning all the test cases and test procedures have to be redone from the beginning. You want to indicate this in the test plan so that all the parties involved agree and understand this is how we're going to perform testing. Test Deliverables. What are the deliverables? What's the documentation that's going to be provided and created during the testing? Could they include the test plan, the test case specification, the test log reports, the test summary reports? Testing asks. What are the different tasks that are going to be performed? You may point to different test case specifications. You may point to different test design specifications. This is where the test plan specifies which activities are going to be performed as part of this testing. Environmental Needs. Where is the test going to be performed? What's the environment? Which software programs are they going to use? Which firmware version? Which facilities? Is it going to be at the agency? Is there going to be a testing lab? What hardware? What power supplies? Which components are being tested? What hardware is needed? Do you need a protocol analyzer? What kind of communications? Do you need a cable, Internet, or Ethernet network? This is the other information related to environmental needs that should go into the test plan.

Patrick Chan: Responsibilities is where you describe the roles and responsibilities for all testing activity participants. For example, the agency will design, prepare, and execute the test. The consultant will manage, review, and witness the test. The vendor will witness the test and repair anomalies. This is a high level. Depending on the risk involved, you may want to provide additional details about each of the activities. Who is going to be there? Who is going to do what? Who is going to approve? Staffing and Training Needs. Sometimes the people involved may require training. For example, the agency representative may require training on how or why we do things a certain way. There may be training or staffing requirements that should be documented in the test plan. Schedule—that's self-explanatory. When are the different testing activities going to be performed? Risk and Contingencies. There's always risk. If you agree ahead of time what the risks and contingencies are, that makes testing activities go a lot smoother. For example, you might be in the middle of testing a specific unit. If a DMS fails, what are we going to do? If a power supply fails, what are we going to do? What if the test units are not available, what are we going to do? Finally, Approvals—the names and titles of all persons that have to approve this test plan. That may include just the agency; it may include the vendors; it may include the consultants who are involved with the testing.

Patrick Chan: We've reached the third activity.

Patrick Chan: The question is: Which of the following information is not being provided in the Test Plan? The answer choices are: A) What item is being tested; B) Who is responsible for performing the test; C) What are the inputs and outputs for the test case specification; D) What are the test deliverables. Again, which of the following information is not provided in the Test Plan?

Patrick Chan: The correct answer is C) What are the inputs and outputs for a test specification? The inputs and outputs for a test case are defined in a test case specification and not in the Test Plan. The Test Plan indicates which item or items are being tested, who is responsible for performing the test, and defines the roles and responsibilities of the persons involved with the testing activities. And what are the test deliverables? Which test documentation, for example, is being delivered by the various parties after completing the performance of each test?

Patrick Chan: The next learning activity is to describe the process of adapting a test plan based on the selected user needs and requirements of an agency.

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: In the remaining slides, we're going to go through an example of creating a test plan and selecting test cases and test procedure specifications. We're going to talk about tools that might be available to help you write test cases for test procedures. And finally, we're going to talk about extensions—how do we test new features that are needed by an agency but are not covered by the standard.

Patrick Chan: The test design specification identifies the features to be covered by the design and its associated tests. It identifies the test cases and test procedures required to perform the testing and specifies the Pass/Fail criteria for testing. For example, we talked about having separate test design specifications—one for blank out signs, or one for a variable message sign. In the next couple of slides, we're going to walk through an example feature to be tested as a specific requirement and how to test the design that the system meets that particular requirement.

Patrick Chan: This is determining which features are to be tested. Recall that the completed standard includes a tool table called the Protocols Requirements List. An agency or a specification writer is expected to fill out the Protocol Requirements List and indicate which features or requirements have been selected for the procurement specification. The Protocol Requirements List was introduced in an earlier module—A311a. This is an example of a completed Protocol Requirement List table where the user needs to activate and display a message. It's mandatory to conform to standards so it's selected. The requirements, "Activate a message" and "Retrieve a message" are also mandatory to conform to standards. Those requirements have also been selected for a procurement specification. "Activate a message for status" is optional, and for a variable message sign or a blank out sign, it's not applicable—so we selected "NA" for this example. But because the requirements for "Activate a Message" and "Retrieve a Message" were selected, and these requirements are mandatory, they should be included as part of our overall testing activities.

Patrick Chan: Using a different example—this is the one we will walk through for the next couple of slides. This example also shows up in the Student Supplement. The Student Supplement will provide more details, including some of the text from the standard and text for the user needs and the requirements. It will provide you a detailed walkthrough of how to create test documentation for this particular feature—to activate pixel testing. In this example, we have a different user need to determine sign error conditions—a high-level diagnostic. It's mandatory for conformance with NTCIP 1203. For the middle one, we have a requirement called "Activate pixel testing," which is mandatory for the matrix—meaning that if it's not a blank out sign and it's not just a pure text sign, then it's a required requirement. The requirement is 3.5.3.1.1.2 and it's been selected in our PRL—Protocol Requirements List.

Patrick Chan: Moving on, now that we've selected the requirement we want to test the requirement. We go to the Requirements Traceability Matrix. This is another tool in the standard that defines the dialogs and data objects that must be used to meet the requirement. This is what the standard specifies. You have to use this sequence of data exchanges and events that are defined by the standard to meet this requirement. Conformance testing confirms that the DMS system performs the same sequence of data exchanges and events as defined in the standard and data object.

Patrick Chan: Looking at the Requirements Traceability Matrix, we see that for "Activate pixel testing," we have to use dialog 4.2.4.2 and Section 4.2.4.2. There's some text that describes what sequence of

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

exchanges and events must be used to meet this requirement according to the standard. In addition, to meet this requirement, we are expected to use data object 5.11.2.4.3—“pixelTestActivation” object.

Patrick Chan: Looking at dialog 4.2.4.2, we see that there is a dialog called “Activating pixel testing.” The three different steps are: “The Management Station shall set pixel test activation .0 to ‘test’,” meaning we shall set this object—pixelTestActivation—to the value of test. The next step: “The Management Station shall repeatedly get this object until it either returns to value of no test or maximum timeout is reached.” A Management Station—let’s say you’re a Traffic Management Center—will keep getting the value of pixel text activation from the dynamic message sign until the value of that object is no test. Or a maximum timeout is reached—meaning after five minutes, I give up. It’s still no test. If the timeout is reached, the DMS is apparently locked and the Management Station shall exit the process—meaning if I still don’t see a test after five minutes, something has happened, so go ahead and exit the process. I used an example of five minutes. The standard doesn’t define what the maximum timeout value is. That’s going to depend on your particular implementation and your communications. For example, you’re using a dialup line for your communications, and you have a really large dynamic message sign, you may need more than five minutes. It may take 15 minutes because of the slow communication and because the sign is so big. On the other hand, if you’re on a high-speed Ethernet network, and it’s a really small sign—let’s say 24 pixels by 40 pixels—you might only need 30 seconds to get all of that information. That’s why the standard does not define a maximum timeout. It’s dependent on your system, your sign, and your communications media. But once the pixel test is completed—let’s say you do get a value of no test—the following objects will have been updated during the pixel test to reflect current conditions. The Management Station may then get, as appropriate, either pixel failure table number rows or any object within the pixel failure table. Those objects tell you specifically which pixels failed. It may come back as no pixels failed—that would be great—but those tables will tell you which pixels failed during the test. This is the standardized dialog and the standardized objects that are used to fulfill the requirement to activate pixel testing.

Patrick Chan: Within NTCIP 1203 v03, we have a third table called the Requirements Test Case Traceability Matrix. Based on the requirements selected in the PRL, an agency can create its own version of a Requirements Test Case Traceability Matrix containing only those selected requirements and the associated test cases. The standard provides a Requirements Test Case Traceability Matrix for all the requirements covered by the standard. But if there are only certain requirements that you’re interested in testing, or if you had new requirements that are not in the standard, you really can create your own Requirements Test Case Traceability Matrix to use for your specific procurement.

Patrick Chan: This tailored Requirements Test Case Traceability Matrix becomes part of your test design specification. So now within your test design specification, you have a table that indicates which requirements are to be tested as part of that test design specification. It also identifies the test cases and test procedures to be performed to verify that your procurement has properly fulfilled those requirements.

Patrick Chan: This is an example of a test design specification. You may have two separate test design specifications—one for blank out signs, one for variable message sign. Features to be tested: You may want to include a copy of the completed PRL— Protocol Requirement List—for each specific item. You may have a PRL specifically for blank out signs. It will have the user needs and requirements for that blank out sign. You may have a separate PRL for the variable message sign. That will indicate the user

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

needs and requirements for that variable message sign. Each test design specification might have slightly different approaches because a blank out sign is pretty straightforward—either the sign is shown on or off. The variable message sign is a little more complex, so the approach might be a little bit more detailed for the variable message sign. Each test design specification may have its own tailored Requirements Test Case and Traceability Matrix because each type of sign will have specific requirements—some will overlap, some will be very different, and each one may have its own Pass/Fail criteria.

Patrick Chan: The next two or three slides will talk a little bit about extensions. While we tried to cover all areas with the standard, there are some requirements or user needs that are not addressed by the standard, either because there was no consensus on how to meet the requirement, or because there might be some features that the working groups had simply not thought about at the time the standard was published. The NTCIP standards allow extensions where an agency can define specific user needs or features and/or requirements that are not supported by the standard. It's permitted, but it's not encouraged—there is no interoperability because now you have a system that may not necessarily operate with another agency or vendor's system. An example I generally use for an extension is called legibility. That's a user need where if for some reason the message on the sign becomes illegible—perhaps because part of the sign is not functioning properly—rather than give a wrong impression or the wrong message, you blank out the sign. However, there are different ways of defining legibility. That's why it was not included in NTCIP 1203 v03. Note that Module A311a discusses extensions in more detail.

Patrick Chan: Let's say you define an extension for your agency's procurement. It's important that the procurer or the specification writer document and clearly define what the user need or feature is. This is what we mean by legibility. Could it be that a certain percentage of the sign is not working? Or is it maybe if more than five pixels are not working then blank out the sign? You have to document that and clearly define what you mean by legibility. That's a feature. Customized requirements—this will satisfy that user need. You may have a requirement that says let's establish how we're going to measure legibility. You may have a separate requirement that defines what we should do in case the message is defined to be illegible. Then you want to define the objects and dialogs to meet each of the customized requirements. These are the steps for setting up the legibility for defining which message should appear in case something is illegible. You want to define the dialogs and the objects for each customized requirement. Test cases should be created for testing each of the customized requirements. How are you going to test it? How are you going to verify that the requirement was fulfilled? And finally, identifying the customized requirements and test cases should be included in your tailored Requirements Test Case Traceability Matrix and the Test Design Specification.

Patrick Chan: Our final topic for this module is a separate tool called a Test Procedure Generator. The Test Procedure Generator is a program that's available from the U.S. DOT that can be used to develop test procedures for requirements in NTCIP standards that have system engineering content. The tool can be used to determine the implementation's conformance to their appropriate NTCIP center-to-field standard—NTCIP 1203 is one of these standards—determine compliance to a project specification, and develop test procedures for extensions. If you have your own agency-defined extension, this tool will help you develop the test procedures for testing that the requirement is met. This does not perform the testing; it's a tool for writing those test procedures. As a note, there's also a feature in the Test Procedure Generator that allows the working groups who developed the NTCIP standards to verify that the standard is complete and traceable. This is the link for downloading the Test Procedure Generator. The link can

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

also be found in the Student Supplement that's included with this module. Over the next couple of slides, we're just going to show some snapshots of the Test Procedure Generator tool so you have an idea of how to use it and what it can be used for.

Patrick Chan: To use the Test Procedure Generator to help generate test procedures, the test developer must first start a new session and load the appropriate standard—in this case, NTCIP 1203 v03—as shown in the figure on the slide. Note that this tool is really helpful for standards that do not already include test cases and test procedures, but it still can be used for NTCIP 1203 v03, which does include test cases and test procedures. We'll demonstrate why briefly.

Patrick Chan: Once you load the standard, the TPG is effectively loading all of the requirements and the standardized design within the standard into the program, which is an extension of Microsoft Word. The tool allows the user to create a set of test procedures for the selected standard.

Patrick Chan: The tool then allows the user to create or customize a test procedure for requirements to be selected by the user. Here, we're ready to click on the button to create a new test procedure.

Patrick Chan: Once you select that, the tool then guides the user in creating the test procedures step-by-step, first by assigning an identifier—in this case test procedure 01.00 “Entering a test procedure description”—and indicating the Pass/Fail criteria for the test procedures. You can see on the screen there are spaces, columns, and boxes for providing the description and for the Pass/Fail criteria.

Patrick Chan: The tool then allows the user to select the requirements that will be verified by that specific test procedure. A test procedure may only verify one requirement, but some test procedures may verify more than one—maybe five—as shown on the slide. We show that the first requirements are selected or checked for this procedure. Once we check “OK,” the Requirements section of the test procedure header is populated with the selected requirements. Although not shown, variables can also be defined and selected by the test procedure. We selected these five requirements and they will show up in the Requirements section in the header.

Patrick Chan: The tool then allows the user to write the test procedures step-by-step. It opens up Windows, showing which dialogs and objects trace to the selected requirements. These dialogs and data objects need to be used to meet the selected requirement, and thus guide the development of the test procedure. The window shows a list of the objects that are applicable and should appear within the test procedures for the selected requirements. Here we have a list of objects—such as DMS sign type and DMS sign technology—that we expect should appear within the test procedure.

Patrick Chan: This slide shows a window allowing the user to edit a step in the test procedure. A set of key words is also provided by the tool to assist the user in writing the test procedures. The set of test procedures can then be saved in a Microsoft Word document by the tool.

Patrick Chan: The Test Procedure Generator has a functionality to retest procedures that already have been created and saved in Microsoft Word, including the existing test procedures and test cases in Annex C for NTCIP 1203 v03. The tool then allows the user to edit the test procedures and select only those test procedures that are applicable to the agency specification. So while Annex C of NTCIP 1203 v03 provides generic test cases and test procedures, an agency can say, “We want to be a little bit more specific for our procurements.” This test tool will allow you to edit those generic test cases and test procedures for your agency-specific specification.

Module 17—Applying Your Test Plan to the NTCIP 12 v03 DMS Standard

Patrick Chan: We've reached our last activity.

Patrick Chan: What is the Requirements to Test Case Traceability Matrix (RTCTM) in a Test Design Specification based upon? What is the basis for your tailored requirements to test case traceability matrix? A) Includes all of the requirements supported by the standard; B) Only includes the requirements selected in the PRL that the test design specification is based upon; C) Includes only those requirements that are mandatory to conform to the standard; D) Includes all of the requirements that are contained in the project specifications. Again, what are the requirements to the Test Case Traceability Matrix and the Test Design Specification based upon?

Patrick Chan: The correct answer is B) The RTCTM is based on the requirements selected. The RTCTM should only list those requirements that are specified in the Test Design Specification. It's not just those requirements that are mandatory to conform to the standard—it includes the optional requirements that are selected for a project specification.

Patrick Chan: To summarize what we've learned in this module. We've gone through why we perform testing, what's included in the Test Plan, and what type of testing we want to undertake for a dynamic message sign. We've gone through the NTCIP 1203 v03 standard and identified the key elements, key components, and tools that are relevant to the test plan. We've given an example of a test plan for a dynamic message sign being procured, and we went through the process of how to adapt test design specifications, select test cases, and test procedures within the standard. We talked about extensions, and we talked about how to use the Test Procedure Generator Tool. These are all tools—information that can help you develop your agency-specific Test Plan and test documentation.

Patrick Chan: With the completion of this module, we've now completed the dynamic message sign curriculum. This started with Module A311a: Understanding User Needs for DMS Systems Based on NTCIP 1203 v03, then Module A311b: Specifying Requirements for DMS Systems. This was Module T311: Applying Your Test Plan to Dynamic Message Signs Based on the NTCIP 1203 DMS Standard v03.

Patrick Chan: I'd like to thank you for completing this module. Please use the feedback link below to provide us with your thoughts and comments about the value of this training. We use this feedback to try to improve not only this module but other modules that are provided in the ITS Professional Capacity Building Program and training. Thank you very much for joining us today.