

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Ken Leonard: ITS standards can make your life easier. Your procurements will go more smoothly and you will 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.

I am Ken Leonard, the Director of the U.S. Department of Transportation, 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 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 your 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'll look forward to hearing your comments and thank you again for participating and we hope you find this module helpful.

Narrator: Throughout the presentation this activity slide will appear indicating there is a multiple-choice pop quiz following this slide. You will use your computer mouse to select your answer. There is only one correct answer and selecting the submit button will record your answer and the clear button will remove your answer if you wish to select another answer. You will receive instant feedback on your answer choice.

This module is T203 part 2 of 2 How to Develop Test Cases for an ITS Standards-based Test Plan Part 2 of 2. Your instructor, Manny Insignares, works at Consensus Systems Technologies as Vice President of Technology and has over 25 years of professional experience in the transportation, system engineering and ITS field. He has significant experience in the development and implementation of ITS standards since 1996, starting with the first version of the Traffic Management Data Dictionary Standards, TMDD.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Manny currently chairs or co-chairs the NTCIP 1213 Electrical And Lightning Management Systems Working Group and is a developer of the upcoming NTCIP 1202 Version 3 Actuated Signal Controller Version 3 and NTCIP 1204 Version 4 Environmental Sensor Station Standards. It is now my pleasure to turn it over to your speaker, Manny.

Manny Insignares: The first slide that we have here is the target audience, which shows the traffic management and engineering staff, maintenance staff, system developers, test personnel, and private and public sector vendors and manufacturers. These are the key stakeholders in a system that works and that's what we're going to be wrapping up today is discussing test cases as they relate to testing because we want a system that works at the end of the day and that's why these key stakeholders are the target audience is important to identify up front.

This slide shows the recommended prerequisites. Essentially it's the curriculum path up to this course module. T101 is An Introduction to ITS Standards Testing. T201, How to Write a Test Plan. T202, Overview of Test Design Specifications, Test Case Specifications, and Test Procedures. T203, Part 1 of 2, How to Develop Test Cases for an ITS Standards-Based Test Plan, Part 1 of 2. That provides more detail related to test case specifications. So this is part 2 of 2. And this is a diagram of the curriculum path.

The learning objectives at the top of—grayed out—are the ones from part 1 of 2. We reviewed the role of test cases. We had a discussion about ITS data structures that are used in NTCIP and center-to-center standards such as TMDD. We learned how to find more information, the information you need to be able to develop your test case and we walked through developing a test case.

In this module, part 2 of 2, we will show how to handle standards that are with and without test documentation. We will show how to develop a requirements to test case traceability matrix. We'll identify types of testing and recognize the purpose of the test log and test anomaly report.

So we did a brief review of part 1 in the previous slide. To recap, we're going to go through a couple of charts in case you've forgotten about the structure of IEEE 829, which identifies test documents and their relationships. Borrowing from the part 1 of 2, we showed that IEEE 829 includes guidance for development of a test plan, test design specification, test case specification, test procedure specification, and test reports including a test log, test anomaly report, and a test report. Just to recap again these are

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

documents that many professional in ITS are using and are familiar with the IEEE notation.

We have a chart over here on the right-hand side that shows relationship of the different test documents and we're just going to walk through it. It looks a little bit scary but it's not. We're just going to walk through it and you'll see how this all goes together. We have a test plan that identifies or describes the overall approach to testing. The test design specification describes which requirements are to be tested and associated test cases. Test case specification identifies the objective of the test, inputs, outcomes, and the conditions for execution of the test. The test procedure specification defines the steps to execute a test. And, in this case, one of the key things to keep in mind is that multiple test cases may reference a single test procedure. So we've seen in practice you spend time, effort, and dollars in developing your test procedures and multiple test cases will be able to be used with the same test procedure. This relationship is a way to reduce the cost of developing test documentation. And lastly, after executing your test or during—rather—during executing your testing, you will keep track of your test log. Any problems encountered will be documented in the test anomaly report, and lastly outcomes of the testing summarized in a test report.

Learning objective number 5, handle standards that are with and without test documentation. And the sub-bullets are to learn how to develop test cases where test documentation is not included in the standard. We take the case study approach to this module. We do three case studies and the first case study relates to NTCIP 1205, which is for CCTV cameras. There is no test documentation in NTCIP 1205. Learn how to handle and develop test cases where test documentation is included in the standard and one such case is the Environmental Sensor Station Standard, NTCIP 1204. We learn how to develop test case where, again, test documentation is not included. We wanted to highlight a center-to-center example and that's case study 3 where we look at an example with TMDD, traffic management data dictionary.

So to recap, the content of the ITS standards, some of the standards have been developed with system engineering process, you're using the system engineering process. And this results in the standard having system engineering content and by that we mean ConOps-user needs, requirements, a protocol requirements list, which marries needs to requirements that's in the NTCIP standards for center-to-field. A requirements traceability matrix that relates requirements to design content. The NRTM, which is needs-to-requirements traceability matrix and that is relates needs-to-requirements in center-to-center standards. And design content. So examples of standards that have the system engineering content are dynamic message sign, DMS, environmental sensor station, and even those not listed here traffic management data dictionary. There are a

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

number of standards that do not or did not go through a system engineering process. So they only have design content in them and one example of that is the CCTV. So we've designed some case studies to walk through how to handle these different circumstances for these standards.

This chart shows a list of the NTCIP standards, starting at the top with 1202, actuated signal controller, and walking down through all the NTCIP standards. And at the bottom we see the TMDD, which is a center-to-center standard. We have a column in the middle that identifies whether the standard has system engineering content. So, for example, as we've said before, the dynamic message sign, 1203 and 1204, Environmental Sensor Station, yes, they do have system engineering content. Whereas right below we see 1205, CCTV and 1206 do not have system engineering content. The column on the right shows whether there is testing content in the standard. And the only standards that have the testing content are 1203, Dynamic Message Sign and Environmental Sensor Stations. So our objective here is to explain how to develop test cases, again, depending on the circumstances that you find your standards documentation in.

So how do we prepare for the test case? We're going to leverage the system engineering process—where it is available—and where it is not available, you will need to develop the system engineering content for your project. In developing test cases, you have to realize that the test documentation in the standards is limited, again, as it relates to test cases and we will show how to use the content that's there effectively. We'll save you quite a bit of time to use what is there and you still have to add a couple, do a little bit of effort, to add in those test cases. If you want to have IEEE 829 conformant test cases and the goal is to do that. In part 1 we walked through developing a test case. And in part 2 we formalize that and we reuse that same process over and over again to show you how to walk through these case studies, again, they are slightly different in what is in the standard, but we will show that the same process can be used.

So the steps here, 1 through 6 that we will be showing how to use in the next case studies. One is identify user needs. Step two will be identify the requirements. Step three will develop test case objective. Step four we will identify design content. Step five we will document the value constraints of the design content to develop the inputs and outputs. And then we will complete the test case, which means to complete the template of a test case, which we introduced in part 1.

So again overview of the case studies we're going to walk through. We have the first case study will be NTCIP 1206 CCTV Camera. It is a center-to-field standard. There is no system engineering content. There is no testing content in this standard. Case study 2

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

will be 1204 Environmental Sensor Station, another center-to-field standard. It contains both system engineering content and testing content. TMDD, Traffic Management Data Dictionary is a center-to-center standard. It did follow the system engineering process. It has system engineering content. It does not have testing content.

Let's begin with case study 1. So for 1205, what are the characteristics of the standard? 1205 is, again, is for CCTV. It is for communicating between a center and the camera, which is in the field. It does not have system engineering content. It does not have test documentation and by that we mean that there is no Annex C test procedures in the standard. So the standards in NTCIP that have test documentation have that documentation described in Annex C and the annex is called test procedures. In developing this case study, we are borrowing from previous modules to help us understand what is the system engineering content we need to define for this project. And the key items are A, module A317a and that's another PCB module, Understanding User Needs for CCTV Systems Based on NTCIP 1205 Standard. A317b, which is understanding requirements for the same. And the standard itself, which has only design content and that design content is described in the standard NTCIP 1205 Object Definitions for CCTV Standard.

So this is a table that was developed in the module A317a. It identifies user needs. In this case, the user need that was developed in the module is 3.0, remote monitoring and requirements were defined also. You can see there at the top the heading 3.3.3 status condition within the device and some sub items, 3.3.3.2 temperature, pressure, washer fluid, and ID generator.

If you haven't looked at the CCTV module or you're not comfortable with developing user needs or want to find out more about that, I recommend that you do go sit through part of it or sit through all of the A317a for user needs and A317b module, it talks about requirements.

This is a table. It's the RTM that was developed, the requirements to traceability matrix where we show requirements and design content. We can see at the top, the top row, that the 3.3.3, the status condition within the device maps to a dialog that's shown there as D.1 generic SNMP GET interface. And the requirements on the left-hand side shown in blue, map to requirements that are in the CCTV standard itself. So for example 3.3.3.2, temperature, maps to 3.7.5 alarm temperature current value, which is in the 1205 standard. So once we identify the design content, we need to look up those values to identify the value constraints on these objects. So what are the value ranges for example of alarm, temperature, current value? So step 2 is identify the requirements.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

In this chart, this slide, I just wanted to show you a little bit about SNMP. It's just a refresher here of how a dialog works with the SNMP GET. The first thing is that an object identifier, referred to as a OID and sometimes pronounced "OID" is used to make requests for data from a device. So we have on the left-hand side the management station. On the right-hand side, we have the CCTV and we will send an object identifier. In this case in brackets, where it says "cctvAlarm 5", that is the object identifier for alarmTempCurrentValue. So I send the OID, the object identifier, and the device will return to me a value. We'll do that again. We will send an object identifier, in this case from the standard "cctvAlarm 8." And we will return alarmPressureHighLowThreshold. Just to let you know in subsequent slide here in a minute, we will show where the CCTV alarm 5, where the OIDs can be found in the standard. That's coming up next. But for right now, I just wanted to give you a feel for how you retrieve values from the device. And this continued with the other variables that are specified in the dialog.

What we've done is taken step 3, develop test case objective and we have taken the same template that we developed in part 1 for what a test case looks like. This is a template that we modeled carefully after the IEEE 829 standard for what a test case shall consist of. And at the top here, we have a unique identifier, TC001 test case 1. We give it a title, which is "Request Status Condition Within the Device, Dialog Verification Positive Test Case." Below it is the objective, which is to verify that the system interface to the device in a positive test we're going to ensure that the system interface implements requirements for the following sequence of objects, the 3.7.5 alarmTemperatureCurrentValue, alarmPressureHighLowThreshold, alarmPressureCurrentValue, alarmWasherFluidHighLowThreshold, alarmWasherCurrentValue, and cctv label Objects. So to go to those objects define a status condition for your device. We continue reading, it says the test case verifies that the data value of the objects requested are within specified ranges. It says the object identifier of each object is the only input required and an output specification is provided to show valid value constraints per the NTCIP 1205 version 01 object definitions. So if you're unfamiliar with this terminology, we're going to go through this step by step, especially that last paragraph about the OID valid value constraints. We're going to step through this carefully.

Here we formally write out a test case output specification. Where we have columns for the data concept ID, the data concept name and the data concept type. So at the top, we have again an identifier for this test case output specification, which is referenced from your test case and the title is status condition within the device. You see it's a simple table that has the same objects that we have been talking about in previous slides and where it says data concept type, each one of these is defined to be a data element.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

If we go to the standard, if we go to NTCIP 1205 version 1 and we look at paragraph 3.7.5, alarmTemperatureCurrentValue, you will see some text that looks like what is inside the little blue box here on this slide. So let's parse through this and see what it is telling us about values. The first thing we see is that this is an OCTET string and it's one OCTET. If you're familiar with SNMP, one OCTET is 8 bits, so this is approximately a bite. The range of values for one bite is 0 to 255. So again the value range as specified one OCTET 8 bits. Value range is from 0 to 255. This is one of those cases where the actual value isn't spelled out, as you can see, the 0 to 255 isn't spelled out, but nonetheless we can determine that by looking at what type of data value this is. Then there's the object identifier at the bottom. There's the two colons and the equals and the object identifier is cctvAlarm 5.

Let's do another one. WasherFluidHighLowThreshold. So again we're just walking through a table that listed out the objects and we're just looking at them one by one. Looking at what the value constraints are. In this case the washer fluid is an OCTET string. There are 2 OCTETS and as we've shown before an OCTET has a value of 0 to 255. However in this case, the standard tells you that because this is a percent, a percent of value, that each OCTET ranges from 0 to 100. So values 101 through 255 would be invalid if they were included in one of these bites. And there we have the object identifier.

So we've looked through the standard. We are now able to fill in the rest of our test case output specification. This looks like the table that we saw previously, which has again the previous test output specification except it's filled in now with the values that we have taken out of the standard. We've placed them in the column that says value domain and so we've looked at the standard and these are the values that are valid and as we do our test cases and do our testing, we want to verify one by one that the values are correct. So this is like a rubric for what the data is that's coming back from the device.

So now we have enough information to fill in the test case. Last time we saw this particular test case, which was early on when we were doing the test case objective, we just filled out this top part, but now we have information to fill out the rest. The object identifier of each object requested is required, so that's the input, the OID. We showed that when we sent the little, showed how SNMP works, we send an OID, object identifier and we get back a value from the device corresponding with that ID. For the outcomes, we show that all data are returned and verified as correct per the object constraints of NTCIP 1205, version 01 and then we list here see test case output specification TC0S test case output specification 001, status condition within the device, positive test case.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

And that was the test case output specification that we filled in. We're referencing it from the test case. We've also filled in the environmental needs since one of the needs is to retrieve temperature based value, there's a note that says when testing for alarmTemperatureCurrentValue, set up is needed to measure the temperature. So that we know that the temperature being sent back from the device is correct. For special procedure requirements, there are no special procedure requirements and there are no inter-case dependencies.

Summary of the CCTV test case. We identified user needs from a PRL, which was developed in module A317a Understanding User Needs for CCTV Systems Based on NTCIP 1205 Standard. So that's another module that describes in detail how to develop the PRL, which is the protocol requirements list, which is where user needs and requirements are related. We identified relevant requirements from the RTM, requirements traceability matrix, and doing that effort had its own module A317b. We identified the dialogs, inputs, and outputs and created a list of the objects from the NTCIP 1205 Standard, the requirements and we built out a test output specification. We identified and documented value constraints for these objects and we developed a test case. In this case, for a standard that does not have any system engineering content in it. So again the onus is on the project to develop the PRL, the RTM where there is no system engineering content.

What did we learn? CCTV standard does not have user needs and the PRL does not have requirements, the RTM, go to A317b module and then using the CCTV design, which is in the standard, we've learned to identify and document value constraints.

We're moving on to case study number 2. Case study 2 is for NTCIP 1204 Version 03 Environmental Sensor Station. Let's review the characteristics of this standard. It is a center-to-field communications standard. Again, it's a center that's communicating with an environmental sensor station that's in the field. The 1204 standard contains system engineering content, so it has a PRL. It has user needs. It has requirements, has design content. The NTCIP 1204 also contains test documentation, which means that it does have an Annex C test procedures and our information source listed here is a 1204 version 03 object definitions.

So let's begin. Step 1 is to identify user needs. What we show here is the protocol requirements list from the 1204 standard and we show user needs. This user need, which has an identifier of 2.5.2.1.2, user needs for wind. Monitor Wind is the user name. And we show the requirements, the functional requirements that satisfy the need are shown

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

here as Retrieve the Wind Data and Required Number of Wind Sensors. So the PRL we have the user needs and the requirements.

In this slide we're showing the RTM, which is the requirements traceability matrix, where we relate requirements to design content and as we did with the CCTV, these are the values that we need to look up for each object to identify what are the valid value constraints. That will let us put together our test case.

In this case we're showing the requirement, 3.5.2.3.2.2 Retrieve Wind Data. At this point we're going to deviate a little bit from the path, you know, step 1, step 2, step 3 that we identified. We're going to take a short break here to discuss the requirements to test case traceability matrix that is in the environmental sensor station standard.

The RTCTM relates requirements to test content. In this case labeled as test cases and we can see that we essentially have a list of requirements on the left-hand side. And on the right-hand side we have, again, Annex C, so all of the test cases, all of the paragraph clauses begin with the letter C and then a paragraph number and then the name of the test case.

You follow through in the standard and we go and look for the retrieve wind data test case and this is what you would find in the standard. At the top it says test case 3.3, the title Retrieve Wind Data. It has a description. It has variables, section of the PRL that relates to it, pass/fail criteria, and then follows through with test procedures.

So having the RTCTM and having the test case and test procedure portions that are defined in Annex C is very useful. First of all the test case material is integrated in the test information is related to the requirements so the effort of having to match test cases with your requirements is done. Once you've selected your requirements for your project, you've selected your design, you trace right through to the objects and you're able to trace through with the RTCTM to your test procedures. The only thing that's missing again from the ESS and the DMS is an, this is what you have to add in your project, are what are the input values and the outcomes. So that is what's missing from the test content. For simple devices like environmental sensor stations, dynamic message signs, what is contained in Annex C is adequate. So essentially you're getting an object, you're retrieving it, you're doing GETs, you're doing some simple Sets of the device, you're doing configuration. Overall the simplified approach works very well. For other standards such as actuated signal controller, traffic management data dictionary, where there are more interdependencies, where the dialogs are more complex, IEEE 829 is a better fit for

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

documenting—for development of the test documentation. And lastly there is no direct translation in between what is contained in Annex C where we spell out NTCIP 8007, there is no direct way to translate what is in Annex C and the IEEE 829 format. So we're going to continue with the steps that we've outlined in the cases of dynamic message sign, environmental sensor station, please use the test documentation, the RTCTM. Those are valuable tools and again we'll work for those simple devices, but we do need to develop test cases. You'll be adding those so you can define inputs and outputs and be able to test your value ranges.

So we continue as we did before in step 3. We developed the test case objective. In this case we assigned an ID, TC0012. You've got a title, Retrieve Wind Data. We identify lists of objects that we are going to be retrieving. We put in language that says that the test case verifies that the data value of the objects are within specified value ranges. Again the OID, the object identifier, is the only input, and we provide an output specification that we'll fill out to show the valid value ranges, the constraints for the list of objects here. Here we have the same template as we've used before for the test case output specification. Where we show the data concept ID and this is right out of the 1204 standard, these numberings, these are the paragraph clauses for these objects, which in the column Data Concept Name. We identify the data concept type. So these are all data elements so we're just retrieving, doing some simple GETs.

So we go to one of these, just like we did before. We're going to the CCTV. We're going to go into the environmental sensor station standard, into the design, and going to look at, in this case, object Wind Sensor Average Speed 5.6.10.4. We can see that it's an integer. It has, this is written here in a format called abstract syntax notation, sometimes referred to ASM.1. We see that this is an integer, it's a number and it's between 0 and 65,535. So we specify it's an integer. We specify the value range and we see that there are additional value constraints that this is an enumerated list of values. We have to look at a reference here, which is the WMO, World Meteorological Organization that has a list of codes that are valid for wind sensor average speed. And lastly we show the OID. WindSensorEntry 4.

We'll do another one, windSensorSituation 5.6.10.10. It's an integer. It's a numerical value. In this case we show, in the standard shows what the valid values are. It's an enumerated list. Other unknown calm, could be light breeze, moderate breeze, etc. and it shows here that we have abbreviated the list just to be able to fit everything on this chart. And the object identifier. This is a windSensorEntry 10.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

So we develop the test case output specification. Again, listing out what we did prior. We put in the concept IDs, the Concept Name, the Concept Type. The last part, which is what we've been looking through the standard is to identify these value constraints and to put them into the table.

One more. So now we have enough information to fill in the rest of the test case template. We show that the inputs are the object identifier of each object requested, is the input. For the outcome we developed an output specification for wind data. We enter an environmental need here that says when testing for average wind speed an artificial wind device is needed to provide the wind for the sensor to measure. So we are going to need to do some setup to be able to then proceed with the test case. We have some special procedural requirements, which are wind simulator setup and that is described here as it shows in the test procedures and there are no inter-case dependencies.

Summary of the ESS case study. We identified the user needs from NTCIP 1204. Those are in the concept of operations of the standard and we can look at those in the PRL. We identified relevant requirements by tracing from the needs in the PRL to requirements. We then used the requirements traceability matrix to trace requirements to relevant design content, dialogs, and the objects. We identified the dialogs, inputs, outputs, created a list of the objects. We identified the value constraints. We developed the test case.

What did we learn? We learned that the NTCIP 1204 Standard has system engineering content, user needs, has requirements (RTM), it has test documentation, the Annex C, and using the ESSS design objects, we've learned to identify and document value constraints so that we can fully develop a test case.

Now we have an activity. Which of the following standards provides testing content? The answer choices are: a) TMDD version 3 center-to-center standard, b) NTCIP 1204 version 3 the ESS standard, c) NTCIP 1205 version 1 CCTV standard, or d) all of the above. Review of the answers. B is correct. The NTCIP environmental sensor station standard has system engineering and testing content. A is incorrect. The TMDD only contains an RTM. An RTM has system engineering content but no test documentation. The CCTV standard has neither system engineering content nor test documentation. So all of the above is incorrect. The only correct answer is B.

Continuing on with case study 3. Case study 3 is for TMDD, traffic management data dictionary, which is a center-to-center standard. Characteristics of TMDD, it has system

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

engineering content, it has an NRTM, which relates needs and requirements, it has an RTM, which relates requirements to design. The TMDD does not contain test documentation. TMDD has two volumes. Volume 1 called ConOps and Requirements and Volume 2 Design. So when we look for our design content, we have to go to volume two of the standard.

This is a little context diagram to show you what we will be developing the test case around. There's a dialog in TMDD called linkStatusRequest where "link" refers to the roadway network; it's a segment of roadway. So we show that we can send from an external center to an owner center linkStatusRequestMsg and upon receipt and verification that you have an authorized requestor, the owner center will send back a linkStatusMsg, which is the response and which contains the data that you're looking for, which is the status of the link. How's my roadway segment doing?

This is a portion of the needs to requirements traceability matrix from TMDD. Again step 1 is identify user needs and because we know this is an NRTM, needs-to-requirements traceability matrix and it relates needs to requirements, we have requirements also listed in this table. These are the requirements that satisfy user needs. So we have user need at the top 2.3.4.2.2. This is need to share link state. It traces to requirement 3.3.4.3.2.1, which is a dialog send link status information upon request.

We can then trace into the RTM, which is the requirements traceability matrix from the needs once we know what our requirements are, we can go to the RTM, which relates requirements to design content and again the design content is in Volume 2 of the standard. So we will go there to look up these values to identify what the value constraints are.

As we've been doing with the other case studies, we will develop a test case objective. In this case we have a TC001 link status request response dialog verification, positive test case. Again, it's to verify system interface implements. Requirements for 1 link status request response dialog and the contents of the link status request message and 3 the contents of the link status information message. So one is the request and one is the response. This test case verifies that the dialog, the request message content, and the response message content are correct. By sending a request message across the system interface and verification that the response message is correct. Input and output specifications are provided to verify the request and response message are correct.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

As we've done before, we can trace from requirements to identify all of the, we're using the term "objects" for center-to-field and center-to-center we were for generally to these information specifications as data concepts and just to refresh your memory, we are also verifying the structure of the data to be correct. In this case we show that this is a data structure. It is a data frame. The first one that comes up is the organizationInformation data frame is one of the design content items that relates to the requirement. This data frame references other data frames and they're listed here, the content details, organization center information, date, time zone and this data frame references other data elements, organization resource identifier, resource name, etcetera. We then show that the structure is correct. This part of the definition shows the sequence of data concepts that make up a correct organization information data frame.

We can drill down a level to find one of the data elements that's specified. For example, link status. Link-status is a data element and only data elements contain values. So when we are looking at the data element, we don't have any additional structure where the...we've traced all the way through the branches of the tree, if you will, we're at the leaves and we have values. So now expect to see a value definition not unlike what we've seen in prior case studies. In this case, the link status is an enumerated list, 1 through 5 and you can see the valid values as being no determination 1, open which is 2, restricted which is a value 3, etcetera, enclosed, and other.

We can define a test case input specification. So this is for the request message. So we list what are the data concepts. In this case we are defining one of the key items here is we're doing a traffic network information request. That's the message. It parses into a data frame, organization requesting, and a couple of data elements. When making a request of the network, we see here the valid values are 1 a node inventory, 2 node status, 4 link status is the one that we want in this case, to be able to request link status. And this is a test case output specification for the link status information that's returned once we make the request.

We can put it all together now in our test case where for the input, we specify a test case input specification that we developed and we specify that we have to set network information type to 4 or the text link status. The outcome, we expect all data are returned and verified as correct. Correct sequence of message exchanges, correct structure of data, correct valid value of data content and we reference a test case output specification. There are no additional environmental needs outside of what is specified in the test plan. There are no special procedure requirements. There are no inter-case dependencies. So we have completed our test case for center-to-center for TMDD.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Summary in this case study. We identified user needs. Went to the NRTM, needs-to-requirements traceability matrix that are in the standard that is in the standard, rather. We identified relevant requirements. We traced from the NRTM in volume one to the requirements. The RTM is in Volume two, which traces requirements to design content in the TMDD, in the dialogs, and all the data concepts. We identified dialogs, inputs, and outputs. We created a list of data concepts from the requirements. We were able to fill out test input and test output specification. We identified and documented value constraints and we filled out input and output specifications. And lastly we were able to develop a test case, completed that, and referenced our input and output specifications to match the language that we have in the test cases.

Summary of learning objective 5. Handle standards that are with and without test documentation. We used a case study approach. We looked at a standard that does not have system engineering content, does not have test documentation, and that was case study 1, NTCIP 1205 CCTV. We learned how to develop test cases where test documentation is included. Also system engineering content is included in this standard and that is NTCIP 1204 environmental sensor station. And in our third case study, we looked at a center-to-center standard that has system engineering content, but it does not have test documentation.

Moving on to learning objective 6, develop a requirements to test case traceability matrix. We will discuss the RTCTM and test coverage. We'll discuss the format of the RTCTM and the importance of testing every requirement at least once. So how does the RTCTM fit in test coverage? Generally the term test coverage indicates the degree to which a test item is covered by test cases. These test items are your test case specification, where we identify requirements that will be verified by your test cases. And this relationship is documented, this relationship of requirements in the test cases that are used to test verify them, that they've been implemented and deployed in the system. You put that in the RTCTM. A simple inspection that all the requirements you intend to test are accounted for in the RTCTM is sufficient. And generally, we want to have each requirement tested at least once using a positive test case and a negative test case to verify that error conditions are handled by the system properly.

Let's look at the format of an RTCTM. In this case this is from the environmental sensor station standard. So we have on the left-hand side, essentially requirements. On the right-hand side, what are the test cases. On the requirement side, we have a requirement ID and a title. In this case, these were taken directly from the PRL. You could develop

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

similar if you have an NRTM. NRTM is very similar to a PRL. PRL is protocol requirements list, relates user needs to requirements in the center-to-field standards and the NRTM is a term we use for the table that relates user needs and requirements in center-to-center standards. On the test case side, we have unique test case ID and the test case title. Each test case verifies whether a stated requirement is implemented and working properly. We will show how each requirement is handled. This is an example. We're looking at requirement 3.5.2.1.9 configure snapshot camera. So this is from the ESS PRL. And we've copied the text. It says upon request, the ESS shall store a textual description of the location to which the camera points and the filename to be used when storing new snapshots. We identified the relevant objects. In this case, this is the design content 5.16.3.1 ESS snapshot camera index. I'm not going to read all the detail here, but the camera description and the camera file name. So we see that that matches what is described in the user need. So we have requirements that satisfy that need. Putting together the RTCTM, the entry, shows that we put 3.5.2.1.9 configure snapshot camera and essentially next to it and below it what is the test case that we will define. In this case, it is defined in the standard. Then we'll define in the RTCTM.

This is a filled out RTCTM. Similar as we saw when we were going through the case study for the environmental sensor station, where we have again requirements on one side and the test cases that are used to verify those requirements in the system on the right.

So now that we know what the RTCTM should look like. We have a template, we have a good example to build from, that we can learn from. Let's build an RTCTM for that CCTV case study that we did. Case study 1, which did not have system engineering content. We have step 1, we're going to identify requirements. We're going to identify test cases that will verify the requirements and that was the purpose of case study 1 in our previous learning objective. So we're going to go back right to that one and then we'll add in an RTCTM entry just like we showed with the ESS a few slides back. We're basically filling in rows to this table, one requirement at a time so that we ensure that all the requirements are in the table and that way we know that we have all positive test cases taken care of.

So we're going to build this. We're borrowing directly from the case study. This is the RTM. We show the requirements and in this case we're going to ignore the design content. So this content is typically, this is from the RTM, you can go from requirements to your design, but we're going to ignore that. We're going to use the same structure here. It makes it easy for you to follow the RTM or the RTCTM. We will build the RTCTM entry.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Let's just as a refresher, look here at the case study, case study 1 for CCTV. We see that we have ID TC001, that's the ID of the test case. We have the title request status condition within the device. This is a dialog verification, positive test case. Then we can put it all together. We have a requirement, an ID and title on the left-hand side and we have the test case on the right-hand side, test case ID, and the test case title. Again, you would go through the process of doing this until all requirements are listed and you can show which test cases are used to verify those requirements. You want to make sure that you identify if there are any special error handling. You want to ensure that you have test cases to invoke every one of the errors that are defined in your standard or in your project so you can verify that the system is handling errors properly.

So what's the importance of this? Every ITS project's testing documentation should include an RTCTM. You must have one. The RTCTM allows testing personnel to focus on each requirement one at a time. The RTCTM brings users, developers, testers on a level field to discuss what a successful outcome is and so you have expectation of essentially how much work it is to verify that you have all the requirements in your system. And without an RTCTM verification, validation cannot be done properly.

So we have an activity. The requirements to test case traceability matrix relates which of the following? Answer choices: a) requirements and design, test cases and requirements, needs and requirements, d) none of the above. And the correct answer is that the RTCTM relates test cases and requirements that the test cases verify. A is incorrect. It is the RTM, the requirements traceability matrix that relates requirements and design content. C is incorrect. It is the PRL or it is the NRTM that relates needs and requirements. And D is incorrect because B is the correct answer.

Summary of learning objective 6. Develop a requirements to test case traceability table. How does the RTCTM fit? We reviewed how the RTCTM fits in the test coverage. We discussed the format of the RTCTM and we discussed the importance of testing every requirement at least once.

Learning objective 7. Identify types of testing. We have a list here of types of testing that we will review. We have a function test, performance test, load test, stress test, benchmark test, integration test, and system acceptance test.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Types of testing. Function test. The function test verifies that the system interprets your inputs and outputs correctly, performs the desired outcome, and returns a correct response. Essentially we're verifying the functions of a system by looking at inputs and outputs. The positive testing invokes a system function to verify that it works properly. The negative test invokes a system error, I'll say on purpose, to verify that the system responds properly to the errors. So a negative testing is used to make sure that your system handles errors properly. We'll define a boundary test. These are constructed to test the inputs and outputs of a system at extremes in terms of value and size ranges. The boundary test is a form of positive and negative testing.

Types of testing. Performance test. Performance test is constructed to verify the performance requirements of your system. For example, those that specify system timing. Testing that verifies that round-trip communications of messages or as we showed before when we send an OID, we expect to get back a value. That message exchanges within the system occur within a specified amount of time. Performance testing verifies completion of a function within a specified amount of time. So, for example, if you have a calculation that has to be done or a—I'll say complex query, but it could be any query that has to be done by the system, you can put a performance requirement on completing that calculation or completing the query and performance testing is used to verify that that function is done within a timely manner. When conducting your performance test, you should give special consideration of how you will verify and log the start and end time of the test. In other words, you have to know when you start and end so you can determine what was the duration of the test. And therefore verify that the performance requirement has been satisfied.

Load test. A load test is constructed to place a demand on the system and to verify and measure its response. The load test is performed to determine how the system behaves under normal and anticipated peak load conditions. So as an example here, you may have a system where you define that you need to have 20 or 40 users accessing the system at the same time. Occasionally, you may go to 60, whatever you define the peak load to be. That is what you want to be able to test the system at to make sure that the system doesn't break when it's at its design peak. Helps to identify the maximum operating capacity of your system, identify where bottlenecks are, and determines which element may be causing a degradation. Special consideration must be given to identify metrics and what you're measuring in terms of systems capacity to handle a load, a demand.

Stress test. This is a type of load test where you measure the system at peak load and overload conditions. So the stress test you define a stress load that is so great that you would expect error conditions to be coming out of the system and essentially the stress

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

test really is intended to try to break your system. You can then assess how well your system recovers from extreme loads and whether the system is able to restart, if that is the intent of when it hits the stress load. It will shut down and restart or whether certain components are supposed to be turned off so that the system can continue functioning. For example, in some degraded mode. These are the kinds of things that are tested in the stress test.

Other types of testing. We have benchmark testing. It can be used to identify that the system achieves some defined level of functionality and performance. So for one example is if we have a number of centers that we're trying to integrate into a region, we may want to define some benchmark that before we integrate any center that they have to have passed a certain level of testing so that they can then share information with other centers. The certification testing, it's a form of benchmark testing, but special attention is given to ensure that systems are treated equally during the test. And a summary benchmark score can be used to rank whatever your results are.

Other testing includes integration tests. We want to test how well system elements work together. So system is a combination of components. We want to ensure that when they work together, the system acts as one thing. It's used to test the operation of the systems, but also to help test what happens if you have a system or many components but let's say you want to test turning off one of the sub-systems or turning off some part to see how the others respond. These are some of the testing you can do during integration tests. Special consideration may be given to document how the system operates under degraded operation. So there may be some functionality within the system that gets turned off, as you turn off some elements of the system, you may go to some degraded mode. So the system can continue to function, potentially doing some critical types of functionality and not shut down altogether. All those are the kinds of things we would test during integration testing.

System acceptance test. Used to verify functions and performance of the system. Typically these are in relationship to some form of milestone, potentially some form of payment. You'll see a system acceptance testing. Typically you will have parties attend to witness and to sign off on the test. Again, disputes may arise at some point during the system once it is in operation, the disputes over payments, disputes over functionality. System acceptance testing is designed to help avoid the disputes, but basically use a testing approach to show that everybody saw what was being done, everyone agreed, and we were able to move on.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

Activity. Which of the following is used to test error handling of a system? Answer choices: system acceptance test; negative test; periodic maintenance test; unit test. Review of answers. Negative test. A negative test is designed to test error handling by the system. Acceptance test is a test not intended specifically for handling errors. Again, we just covered that. It's more for helping you—all the parties agree to a milestone or payment. C periodic maintenance test, again not intended to handle a particular error. It's just something that is done when the system is operational and you may periodically want to do some maintenance testing to see that everything is working as you would expect. Unit test is a kind of function test of a subsystem or unit of the system.

Summary of the learning objective 7. Identify types of testing. We discussed the function test, positive test, negative test, and boundary test, performance test, the load test, stress testing to break the system, benchmark tests, the integration test, and system acceptance test.

Learning objective 8. Recognize the purpose of the test log and test anomaly reports. These are outlined in IEEE 829. We're going to discuss the test log, identify data, information, files, signatures needed. We'll discuss the test anomaly report. Identify what failed, what investigation is necessary to provide feedback to system developers and maintainers. More in-depth discussion is provided in module T204 How to Develop Test Procedures for an ITS Standards-based Test Plan. And that is during your testing and your test procedures, you'll identify what needs to be logged in and the incident reports, the anomaly reports that have to be written up whenever there is an incident or problem that's discovered during testing. So these are covered in more depth in T204.

So impacts of failure of a test case. So we've talked about this here because we want to relate what you do when your test case doesn't work. So we've come up to talk quite a bit about test cases and development, but what do you do when your test case fails? Ideally the test plan will include language to describe what investigation should take place in the event of a failure or error during testing. So in the event of a failure that relates to a test case, you must fill out a test anomaly report. In many cases test engineers or developers of the system being tested are present who will know what the error stems from. They can describe what the problem is. They may be able to quickly isolate the problem right then and there and make a note about it and depending on the test engineer, you may want to try again. Data is something that's easy to identify. If it's something that requires more investigation then you document as much as you can about the situation that led to the problem.

T203 How to Develop Test Cases for an ITS Standards-Based Test Plan Part 2 of 2

The test log. You will maintain a test log to provide a chronological record of relevant details relating to execution of the tests. You'll capture data, files used, locations of information, locations of devices potentially, the direction they were facing if that is critical part of your environment. Date and times of execution of the tests. And there's a test plan that should spell out the detail of what should be logged.

The test anomaly report will document any event that occurs during the testing process that requires investigation. So again, you're going through your test procedure where you will track and identify what is pass or fail. When something fails then you need to create a test anomaly report. And just a note here in the top bullet that a previous version of IEEE 829 this was called an incident report. And the key thing is to identify is what the problem was, defect, trouble, any issues, etcetera.

Summary of learning objective 8. Understand the purpose of the test log and the test anomaly report. These are outlined in IEEE 829. We reviewed what is the purpose of the test log. We reviewed the purpose of a test anomaly report and more in-depth discussion is in T204, module T204.

What have we learned? We showed that some standards have test documentation, some do not, and then how to deal with these gaps. We showed elements of a requirements to test case traceability matrix. How it is constructed. How to account for all the requirements and all the test cases. We learned how to document and handle the impact of test cases. What is a test log, what is a test anomaly report. We learned about types of testing.

What have we learned? There's a process to gather information and develop a test case. Identify user needs. We identified requirements, developed a test case objective, identified the design content, documented the value constraints in your input and output specifications, and we completed the test case. We've learned about test reports for documenting test case failure. Document in the test log and the test anomaly report contains the detail. Test cases can be re-used in different types of testing. We reviewed function test, performance test, load tests, stress test, benchmark test, integration test, and system acceptance test.

We have a list of resources you can go to. We also have a student supplement. We've combined parts 1 and parts 2 into a single student supplement. I want to thank you all for attending this online training.