Module 5 - T101

Introduction to ITS Standards Testing

Webinar Transcript

(Note: This document has been converted from the transcript to 508-compliant HTML. The formatting has been adjusted for 508 compliance, but all the original text content is included.)

Shelley Row:

ITS standards can make your life easier and you'll encourage competition but only if you know how to write them into your specifications and test for them. This module is one in a series that covers practical applications for acquiring and testing standards based ITS systems. I'm Shelley Row the director of the ITS joint program office for US DOT and I want to welcome you to our newly well designed ITS standards training program of which this module is a part. We are pleased to be working with our partners 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 would tell colleagues and customers about the latest ITS standards and encourage them to take advantage of the 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 which improves livability for us all. You can find information on additional modules and training programs on our website: www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through our evaluation process. We look forward to hearing your comments. Thank you again for participating and we hope you find this module to be helpful.

Nicola Tavares:

Today's webinar is T101 introduction to ITS Standards Testing. The target audience are engineering staff, operational staff, maintenance staff and decision makers. Your instructor is Dr. Gary Thomas, he is a research engineer with the Texas transportation institute, and he has a bachelor's degree in civil engineering from the University of Minnesota and a Masters and Doctoral degree in civil engineering from Arizona State University. He has many years of experience teaching ITS standards.

Gary Thomas:

Alright, thank you Nicola, hopefully everybody can hear me alright. Glad to have you here for this webinar T101 which is introduction to ITS standards testing and this is going to be a very broad level over view of what testing is all about. The course will provide you with an introduction to the need for, and the applicability of ITS standards testing and the concepts related to conformance requirements to design, documents to be merged, systems tests, and interface testing will be discussed. There are some prerequisites here on the screen and hopefully you've been able to take part in them before today. This is really the beginning of another curriculum path in the ITS standards PCB program.

So you can see on screen we're at T101 right now the introduction. After this introduction you're encouraged to go on to some of the other courses which either have been developed or are in the planning stages of being developed and they will go into much more detail than we will go into today. Especially *T201 How to Write a Test Plan,* that gets into much more very specific details on testing and how to go about doing that. So today we're going to stick to much more of a higher level introduction to this topic. The learning objectives today by the time we're done you should be able to explain the need for and benefits of testing. And while I'm going over some of these I'm going to ask you in just a few minutes to chat in what you think some benefits of testing are. So I want you to kind of start thinking about that right now and when the time comes on that slide I will prompt you to go ahead and put your comments in the chat window. I'm going to ask you about benefits of testing.

The second objective is to describe how ITS standards fits into the overall scope of a systems test and a systems life cycle, especially as it relates to the systems engineering process. Third you should be able to discuss how to test an implementation for conformance to standards, and finally you will be able to distinguish the difference between standard conformance and project compliance. First of all I want to talk about testing that's something of an ambiguous term that's being used especially when it comes to standards. There's really two types of testing which we're only gonna cover one today. One type is standards testing which provides information to users on the reliability, interoperability, functionality and performance of systems using the standard. That's not what we're talking about today. The second one is systems conformance to documented test plan based upon an ITS standard. That really is the topic of today's webinar. So as I mentioned this is our first chance to chat something in and what I want you to do and I see one has already come in. Tell me what you see as possible benefits of testing so you can use the chat pod to chat that in.

So the first one came in thank you is ensures that the system will operate as expected. That's a very good point as you do testing and we'll talk about it in much more detail today when you put out a test plan that's really telling the contractor or the system's integrator what you expect of this system. So by passing tests we know that it will operate as expected. Another one is avoid costly surprises yeah that's a good point yeah you don't want to come down the road a little while and if something hasn't passed a test if you didn't do testing. You can also get to the implementation stage and you might be surprised that something is not working as it should be. Another one says determine if system needs are requirements. Exactly you'll establish some requirements and the system needs to meet it and you do that through different types of testing. To chart achievements during and after installation, that's a good point as well. We'll talk about that a little bit where you can actually have some benchmarks throughout the actual implementation of a new system using ITS standards and so we can document that during the process and after installation as well. Another one; become familiar with new system. Yeah as we do certain types of testing your staff can be involved in the testing process before the system actually gets implemented for real. Requirements verification, yep we can do that and also better quality is the last one chatted in. By testing to a certain standard we know that we can achieve – we're going to test to a certain amount of quality that we want our system to perform. Very good thank you all for chatting in those benefits of testing those are all very good answers.

I can tell we've got some experts in the audience today. Well here's some that I've got up here and there are many benefits as I've pointed out and I'll just kind of go over these and some of these have already been mentioned. So verify that requirements are fulfilled. I think that was one of them that mentioned to determine if the system meets our requirements, exactly. To reduce the risk of misinterpretation between agency and manufacturers. That's another benefit of testing because we all know when we have contract documents and especially as the project gets even bigger and bigger in scope there's certainly more opportunity for misinterpretation between the two. Reduce the risk of financial mismanagement because testing provides clear benchmarks as I mentioned earlier for a contractor to demonstrate that the device or system is working as designed or intended and you can also structure your payment schedule on passing those tests so that's why you can reduce the risk of financial mismanagement. Also reducing the risk of perceived lack of oversight and while most projects are closely overseen by those doing the implementation doing testing certainly provides some solid documentation to any outside parties that may be critical of a large project.

Gary Thomas:

And like me even small projects and large projects are often times criticized by those outside looking for something to be wrong with the project and testing can help that out. And finally ensuring interoperability to allow for system expansion. So as you can see three of these benefits really relate to reducing risk and risk is going to be discussed in a little bit more detail in a later slide. And was pointed out by another person even requirements can be misinterpreted and it's best to have a dialog to ensure mutual understanding. Absolutely because you've defined a requirement doesn't mean it can't still be misinterpreted and by doing our testing and we'll talk about this shortly you can have some good hard numbers which hopefully are not as misinterpreted. Well, with any ITS testing including your center to field, center to center, and device testing the agency really needs to identify the party that conducts the testing as well as appropriate performance provisions and have a clear understanding of the responsibilities of all parties involved in the testing. And the agency should also include a description of the consequences of test failure in terms of procurement, budget schedule, things like that.

So who can test? Who are the testing parties? First of all the agency may do the testing. The agency testing is that component of the testing framework that's carried out by members of the agency staff. It's expected that the agency possibly with an outside consultant possess the technical knowledge, the tools, the staff, to conduct this basic testing independent of manufacturer assistance so that's really going to depend on the expertise you have in house. If you're a smaller agency that does not necessarily have a lot of expertise in standards or testing to standards you may need to hire that out to a sub consultant to handle those duties for you. If you're a large agency you might have the in house expertise or you still may want to team up with somebody. Also the manufacturer covers those tests that are conducted by the manufacturer obviously. Procurement of a field proven off the shelf ITS device that may not necessitate additional manufacturing testing.

But if an ITS device is being developed specifically for the agency or an existing ITS device is being modified for the agency then additional tests should be required. And the agency or its consultant should witness such additional manufacturer tests. And finally another party that could do testing is independent laboratories. Independent laboratory testing is often used to verify that an ITS device meets certain environmental specifications such as vibration, temperature conditions, power conditions, and so perhaps light output. And such testing is often required for traffic controllers and other ITS devices, and a lab will furnish the facilities, certify for accuracy typically. And the personnel capable of performing the tests and then will record the outcome. However they typically do not prepare test plans to do that. I think of an analogy for the independent laboratory is things like electrical power where something might be certified by Underwriter's Laboratory, UL it's a UL listing that you would find. You can see that on just about any piece of equipment that you buy for your home that's electrical in nature. You'll find somewhere on there a little label that says UL listed. In that case an independent laboratory has confirmed that it meets those particular requirements for plugging it into the wall.

As I mentioned all parties should have a clear understanding of their testing responsibilities and that really comes up in the test plan which we're not gonna cover in a lot of detail today because that's the next course in the curriculum path but you will really define that in the test plan. And also as I mentioned earlier having a clear description of the test failures, what do you do if you have a test failure, how much time do you have to fix it in order so it does pass the test? Things like that? Well again I'm looking at terminology. A lot of it can be somewhat confusing terminology when it comes to testing. And what I like to talk about here for just a minute is two items called conformance, and compliance it's important to understand the distinction of the terms so I'll go over them here.

Conformance is a condition that exists, so it's a condition that exists when an item meets all the mandatory and optional requirements as defined by a standard. A standard contains a conformance statement that defines the necessary conditions for a product to claim conformance. Compliance on the other hand, again it's a condition that exists when an item meets all of the requirements of an agency specification and I'm gonna go over there. I'm gonna revisit these in a little more detail later in the webinar. A few more terms I will talk about here are validation, verification, and traceability. Now validation is making sure that a system when placed in operation will support agency needs. In other words it kind of asks the question have we built the right thing. So in your process – and we'll go over this also in a little bit the systems engineering process you set out certain needs that your agency requires and you used validation to figure out if indeed what you have built meets those. And that's answering the question have we built the right thing? Verification is making sure a design complies with requirements and the systems as both proposed and delivered, comply with by design and requirements. So here we're answering the question have we built the thing right? So it's kind of a subtle difference there verification is looking at your requirements and making sure that it, whatever system you're implementing tests to that. And traceability is this key tool in validation and verification, traceability helps to determine that the systems – sorry if the agency's requirements were fulfilled by the system design and that implementation was done correctly by the installer. So where does that bring us? That brings us to testing as it relates to the systems engineering process. Now hopefully you've probably taken some previous webinar perhaps in systems engineering that's one of the introductory level courses in this PCB program is the systems engineering process and how it relates to the ITS standards program. And so here we have the system's engineering process. I'm not gonna go over it in detail here but as we see if we're coming up the right hand side of the V, of the diagram that's where we get into our testing, and our verification, and validation and you can see those words on there. And we're gonna go over each of these in a little bit more detail.

And traceability is how we connect the descending side of the V to the ascending side of the V, traceability going back to what we set out in our concept of operations. What our system requirements are and in the design, the high level design and the detailed design. So testing is an important aspect of the systems engineering model obviously and before any testing begins there must be clear statement of understanding of the requirements that must be met and the minimum accepted performance levels. Then all testing should be based upon and derived only from these agreed upon requirements. Each requirement has a test and each test traces to a requirement. And again if you have questions at any time you can chat those in on the chat window.

Gary Thomas:

Alright testing – still talking more about testing in the systems engineering process. Unit device testing focuses on comparing an implementation against the standards and specified options; this may be performed by inspecting the code to use proven software to send test messages to the device. And this software and hardware components are thoroughly tested to identify as many defects as possible. The first line of defense is the software developed who should step through and test every line of code including all exception and error cases. Additionally a series of test cases are devised that will exercise the hardware, software component. These test cases are documented in a unit verification plan after the software is complete and thoroughly debugged by the developed the test cases are used to test the hardware, software and the results are documented. Identified defects are analyzed or corrected and testing is repeated until all known defects are either fixed or otherwise resolved. The defect correct may be relatively simple it may include redesign the sections of code that are determined to be error prone so that's where this comes in at the very beginning of the ascending side of the V.

Subsystem testing now we're gonna go up a little higher and we're getting into some verification here. Subsystem testing consists of connecting two or more devices together and exchanging data and this is where a lot of the ITS communication standards come into play, where they're exchanging data between either two devices or it might be between the center and the device, something like that. A subsystem testing assumes that the individual devices and subsystem components have previously passed a sufficenctly designed unit test plan. And that the devices or subsystem components support the same ITS operational, and or functional features. Every requirement is verified used the test system – sorry in the verification procedures and system requirements and the related subsystem and the component level requirements may be verified several times as verification progresses. Now for example a requirement say that the system shall blank a selected dynamic message sign on user command might be verified at several different levels, the capability of the sign to blank itself might be verified and the dynamic message sign, DMS component level.

Gary Thomas:

The capability of the user interface to accept and relay a blank sign command might be tested at the subsystem level. And finally an end to end system test would be used to verify that the sign actually blanks on user command. The fully integrated system should be verified at the integrators facility before it is installed at the customer's site. And now looking at subsystem – sorry system testing as we move one more up system verification and system integration. System verification testing is the highest test level it is also usually the one with the fewest requirements remaining to be verified. Only those requirements relating to subsystem interactions quantity of field devices, external interfaces, system performance should remained to be formally verified at this point. Subsystem acceptance testing is performed after all lower level testing has been successfully completed. It's performed in the operational environment using all available and previously installed and tested system hardware and software. The system acceptance test should include an end to end or operational readiness test of sufficient duration to verify all operational aspects and functionality under actual operating conditions. Formal acceptance of – at the subsystem or system level may trigger the start of equipment warranty periods or software licenses agreements, maybe even operational or maintenance agreements. And the procurement documents should clearly specify which of these are applicable. When the become effective and when they need to be renewed and when the payment schedules are. Excuse me for one second here.

Alright I'm gonna talk briefly about an introduction, actually I wanna look at the chat window there a few things have been mentioned let's get caught up there. Is validation testing relevant perhaps there might be a typo in there. Is validation testing relevant to standards testing in particular? Validation testing relevant to standards testing in particular, no, if I'm understanding the question correctly. You can validate on other things other than ITS standards but it's not strictly to ITS standards, if I didn't quite get your question answered there just let me know. Another point is are you going to go into more detail about test cases, what they are, examples on how to write them and how they compare to test procedures. I will go into a little bit about that here coming up but again not very detailed because that's really for the next webinar after this I think it's I201 or 102, I can't remember. **(T201)** that goes into how to write a test plan. So that's going to get into the much more detailed stuff there. The difference between system testing and acceptance testing?

I guess we're talking over here I mentioned I think I mentioned that it includes I have it there under system testing it says includes acceptance testing and I guess you wanna know the difference between those two. The acceptance testing is really geared towards I think that part where I mentioned about formal acceptance at this point might trigger some other things. So you can do some systems testing and at some point you say okay if it passed this test we accept these devices and they are now, the clock is now on for their warranty period. That's the way I would differentiate the two I hope that answers your question.

Alright I'm going to talk a little bit about IEEE 829 which is a standard of testing. IEEE 829 is an institute of electrical and electronics engineer's standard, nobody every spells that out so it's just IEEE. A standard that specifies the form for a set of documents for use in 8 stages of software testing. 8 stages potentially produces its own separate type of document. The standard specifies the format of these documents but does not stipulate whether they all must be produced, nor does it include any criteria regarding adequate content for these documents. These are a matter of judgment outside purview of this standard. Now IEEE 829 is a standard for software testing which involves a lot of testing for ITS type systems. However most the steps and procedures outlined in IEEE 829 are directly applicable to other types of testing, interface and hardware for example. Two of the items included in the standard that I am going to discuss in a little bit more detail, again not great detail but a little bit, test plans and test procedures, the IEEE 829 really provides a framework for testing within the system's life cycle or system's engineering process. Well I'm gonna ask you another question that I want you to chat in some answers to. What do you think a test plan should cover? So what do you think a test plan should cover so chat in some of those and we'll see what you come up with and then we'll go into a little more detail?

Anybody want to volunteer what they think a test plan should cover. What you're going to test? How you're going to test? And where you're going to test? That's great it's practically my next slide. All operational aspects of the specification, yeah. All procedures, test data, environment, outcomes. The procedures, standards, and when the test is successful. Test environment and architecture. All good answers, very good answers, we'll talk about it in just a little bit. So some of the testing considerations, an agency's approach to testing any deployment should consider the maturity of the ITS device software interface. Oh sorry about that hopefully that got rid of that, the maturity of the device, the software, and the interface, the number of units being required and installed. The agency's ability to test, what expertise you have available. The relative significance of each agency specification requirement among other factors. Your approach to testing should really be tempered with number of units to be acquired, the unique agency or project implementation requirements and the history of the device and the NTCIP standards, maybe how matured they are. So there's a lot of different considerations to be taken into account. Somebody else had mentioned if the system worked the way it was designed. Yeah absolutely.

A test plan, so a test plan is a management planning document and it shows, and this gets to that first answer that got chatted in. Who will do it, what will be tested, how it will be tested, when will it be tested and how long should it take, and what the test coverage should be? What quality level is required? With any standard or specification discussion eventually turns to the discussion how do we know if an implementation or an application conforms to the standard and complies with my specification. Well a test plan defines how one answers this question. The overall plan should convey the scope, the approach, resources needed, schedule of testing activities. It is used to present a high level view of the project to inform all interested parties. As defined in IEEE 829 a test plan defines the scope, resources, and schedule of testing activities. It identifies the items to be tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each test and the risks associated. Some more people chatting in, timeline, environmental test, tool test, yep format all those things should be included. What are the benefits of the test plan we've talked about some of these already. One of the benefits is that it provides a framework and a process to verify that the system meets user needs. You can also improve stakeholder participation because we wanna get a lot of other people involved in the test. It is more adaptable, you get more adaptable more resilient systems by having a good test plan in place. You get to verify functionality and you end up with fewer defects on devices. You get a higher level of reuse from one project to the next and it also results in better documentation. So I know I've already said it, I'm not gonna go into any more detail on test plans today. That's the topic of the next module that addresses how to write a test plan.

A question got chatted in does a test plan include test cases? In other words higher level procedures? Yes it does. And here is something a little bit more on test procedures, a test procedure specification details how to run each test. Including any setup preconditions and the steps that need to be followed, so I may not of quite said that on the last slide there slightly different test procedures and test plans. But they're very closely related. So given the sophistication of today's systems and components it's pretty much impossible to test all possible combinations and permutations of how a system or component might react. In general the principal is to define at least one test procedure of each agency specification requirement and exercise those features in the context of how they should react in a normal system operation environment. The test procedures should be written with some flexibility so that a suspected anomaly can be verified perhaps retested as the test progresses. And a format for developing test procedures is now included in NTICP 8007. The test procedure should reference the agency specification requirements and those requirements contained in the NTCIP Standard. So procedures step by step, keystroke by keystroke, expected results. I think that's really gonna depend on again on a number of those things that I pointed out. If the device is a fairly mature device that has been around the block for a while you're probably not gonna go into as much detail as you might for a brand new piece of equipment that does something completely different that what passed equipment has done.

I remember years ago back in the let's see probably be the early '90s we had 8 phase controllers and that was the most controller signals you could have on a traffic controller 8 phases. But where I was working, the state DOT had need for, radical idea, a 16 phase controller, didn't exist at the time. And so the state agency put together a specification for a 16 phase controller and they probably had a test plan that would have gone into much more detail of how this thing should operate and probably got into. I don't know necessarily key stroke by key stroke, but they certainly had to go into much greater detail on how this device was gonna work because it didn't exist before they developed it.

Another question chatted in, how can you test under full load? For example maximum number of users? That's a good question, full load, how would you test under full load maximum number of users. I don't know I'm gonna have to – well we had somebody who chatted in here, there are devices designed to do that. So you can simulate N number of devices using test, not certain what that last word is supposed to be. Test stubs okay test stubs. Alright. So there's a couple of ways you can do it and I do thank those of you who are answering questions. I appreciate it as well, please feel free to chime in on anything that I say or clarify anything that I say.

The testing is generally divided into these phases which I'm gonna go into more detail here on the subsequent slides. I'm gonna go over these phases and it should be noted that the beginning types are more related to hardware testing and not necessarily software interface testing, that type of testing is typically done in the latter phases of the systems engineering cycle. But I do want to give you an overall picture so you can see where it fits in to the systems engineering process. The lowest type, or the beginning type is prototype test and inspection, and the purpose of a prototype test is to verify the electrical-mechanical design of the device for the environmental conditions or specifications. If the ITS device is a custom development much like the 16 phase traffic controller that I was mentioned, or maybe a customer extension to an existing ITS device designed to meet a unique agency specific requirement. Then it may be necessary to identify a prototype testing phase for the device. During the prototype testing phase the device manufacturer constructs a fully operational version of it for inspection and testing by the agency. It typically occurs during the unit device testing phase of the SE process, SE systems engineering process. So we're way down on the bottom of the ascending side of the V diagram. Likewise if you were purchasing off the shelf equipment that is not something new you probably don't need to go into this particular type of testing. They're not gonna build you a prototype of something that already exists of the shelf.

Next phase and again visualize moving up the right side of the V diagram is design approval test and inspection. The second phase of the ITS testing occurs when it's in its final production form. This is generally considered design approval testing and it includes a repeat of the prototype testing and inspection along with shock and vibration testing if it was a particular device that was gonna be out in the field. Design approval testing verifies that the ITS testing device in its production form fully complies with all of the standards of conformance and the agency specifications, compliance. At this point the documentation of the device should be included in the inspection. It typically occurs during the unit device testing stage of the process. Again of the shelf type device probably not gonna go into great detail about those, but the more unique a device is you will probably do that kind of testing.

Factory acceptance test is the next phase that we usually test in this is usually the final test performed prior to the delivery of the device to the agency. And may be observed by the agency, the purpose of the testing is to ensure that the production units have been constructed in a manner identical to the unit that was subjected to the design approval testing, and that all connections and functions are fully operational. Where the device includes analog functions these need to be verified to ensure that the operation is within the acceptable tolerances. This again typically occurs during that unit device testing stage of the systems engineering process. The next phase is the incoming device test. Once the device has been delivered to the agency it should be inspected for damage during shipping and handling. And where possible the device should be powered and run through a test that exercises all the connections, features, and functions, again typically occurring still at unit device testing stage of the process. This phase is the likely point at which an agency would want to do its own independent standards conformance testing. The software interface will be tested against the ITS standards to make sure it is in conformance with the specifications. Testing at this stage will make sure that the devices are interchangeable. So now we're really getting into more of the verification testing that you would have.

So the site acceptance test, once the device has been installed in its intended location for operation, the full ITS device functionality is again tested and verified. If the system is in place this means connecting the ITS device to its complete system and verifying all the operation and feedback of the ITS device. Where the central system and communication infrastructure may not be available this testing should be performed with a standalone central simulator perhaps or a similar tester. Typically occurs during the subsystem testing of the SE process. At this point the testing agency will typically test interfaces between systems that interoperability that we often talk about in your systems.

A see questions coming in I’m gonna go over this last one and then I’m gonna go back to that question. The burn in and observation test. Once the ITS device is installed and verified it is connected to the system and undergoes a period of observation, and this is typically a period of perhaps 30-60 days during which the ITS device should exhibit fault free operation. During this period of time the ITS system should be exercised to monitor the functionality of the unit. In other words put it through its paces another way of saying that. This may include test messages operator use of the camera if that’s the particular device. The operation of traffic signals, data collection devices, those are all examples of what you might be testing. Typically occurs during the systems testing stage of the systems engineering process. So we’ve worked out way up the ascending side of the V to make sure that you know one, that have we built the right thing, and have we built the thing right, answering those two questions.

Back to the question, factory acceptance test is a different from systems acceptance test? It seems to be geared more towards widgets, devices, cameras, etc… versus central systems, ITS hardware management. Yes. Some of those early ones are geared more towards widgets and devices. You’re not gonna do a lot of interoperability testing at that point. As we work our way up to the system acceptance test that we look more at the interfaces and we look more as you mentioned at your ATMS or your arterial management systems. Another question is does burn in intended to be a partial cut over, in other words old system is still in place, and new system is only controlling a portion of field devices. It can be it doesn’t have to be. A lot of times if dynamic message signs are first going in you would probably do a burn in testing of the new system. If you are replacing your ITS devices your burn in – if it’s an already operational system you might do your burn in back at your agency location, you would have it installed there, that would be one way of doing it. You could certainly do it out there in the field if you’re replacing, if you’re taking down one sign and putting up another you would have to burn it in there. But if you’re gonna put up a new structure I guess you could do a burn in separately. I think it all depends again on what type of devices we’re talking about, how difficult it is to burn them in in the shop if you will, you’re maintenance yard, and how the system would be viewed if you burned in a DMS out in the field and you had a system working for a number of years. That might confuse the public to all of a sudden see signs with test messages, so that could cause some potential confusion there. Alright let’s talk about methods of testing one second please.

Now testing is just one method of verification, we’re gonna go into a little bit more detail here. All testing falls into one of these categories, it might be inspection, analysis, demonstration, or formal tasks. We verify using one of these four methods. From verification standpoint inspections are the least rigorous method followed by analysis, demonstration, and finally formal tests most rigorous method. And I’m gonna go over each of these in a little bit more detail. Inspection is the verification by physical and visual examinations of the item. Reviewing descriptive documentation, comparing the appropriate characteristics with all the reference standards to determine compliance with the requirements. Some examples of this might be measuring cabinet sizes matching paint color samples, observing printed circuit boards to inspect component mounting and construction techniques. So again this is not looking really much at ITS standards I only bring it up for completeness of the various types of testing, but you’re not really testing to a standard here, and ITS standard that will come in a later method. But you still do inspection when you get your devices. Analysis is the verification by evaluation or simulation. Perhaps using mathematical representations, charts, graphs, diagrams, calculation or data reduction now this includes analysis of independent of computer implementation analytical conclusions drawn from test data and extension of test data to untested conditions.

This is often used to extrapolate past performance which was accurately measured to a scaled up deployment an example of this type of testing would be internal temperature gradients for dynamic message sign. It's unlikely that the whole sign would be subjected to testing within a chamber. So a smaller sample is tested and results can be extrapolated into the final product. Along the same lines trying to determine the temperature rise within a dynamic message sign may require analysis based on airflow, fan ratings, vent size, all kinds of stuff like that. Other examples include the review of power supply designs to verify that they comply with junction temperature, voltage limitations contained within the procurement specifications. Again not really getting into testing against any type of ITS standards within that particular method. Whereas we get up into these demonstrational informal testing you'll see how it applies a little bit more. Demonstration is the functional verification that a specification requirement is met by observing the qualitative results of an operations or exercise performed under a specific condition. This includes content and accuracy of displays, perhaps comparison of system outputs with independent drive test cases. And maybe system recovery from some type of induced failure condition.

Formal testing is the verification that a specification requirement has been met by measuring, recording, evaluating qualitative and quantitative data obtained during the controlled exercises under all appropriate conditions using real and simulated stimulus. This includes verification of system performance, system functionality and correct data distribution, and this is the type of testing that is certainly most applicable to testing against ITS standards. For some standards such as communication infrastructure and their associated protocols there exists test equipment that can be used. And therefore the test procedure should be instructive on what test equipment should be required by the agency. So formal testing you can use, here is where you are testing the interfaces and how it reacts, how these two devices are talking to each other. Is the correct data being passed from one device to another device or from a central computer to a device? And that's where this formal testing comes into play against ITS standards. Well I told you we'd come back to conformance and compliance in a little bit more detail so we'll talk about that here for a minute.

As I already pointed out conformance is a condition that exists when an item fulfills all the mandatory requirements defined by a standard and any optional portions of the standard that the agency selects. And you'll see this in your NTCIP standards. Conformance testing is designed to verify that the ITS device fulfills the mandatory and optional requirements of the ITS standard. Hence for starting testing one should first ensure that a set of well-defined requirements exist and can be verified. Requirements are the basis for testing and verification. Conformance testing addresses all parts of the standards that are mandatory for the ITS device. Whether that part of the standard is required as part of the agency's specification or not. Conformance to the standard is tested against all mandatory requirements of the standard and some of the optional is written but the standards authority.

So compliance again a condition that exists when an item meets all requirements of an agency's specification. Have we built the thing right? Compliance testing is designed to verify that the ITS device fulfills the requirements of an agency's specification. Creating a specification is necessary to ensure that all optional elements in the standard that are required for your deployment. Specifications also allow for the explicit removal of operational elements that do not apply. Therefor while a manufacturer may state that an ITS device is NTCIP conformant or compliant, sorry NTCIP compliant, meeting conformance with the mandatory requirements of the standard might not be sufficient to fulfill your agency's project requirements. Or to comply with your agency's specifications. Compliance as tested against the requirements developed is part of the project. Within NTCIP conformance is a subset of compliance given that the requirements are selective on the basis of user needs selected by the agency. Now I'm gonna ask you another question here.

Based on that why do you think conformance is necessary? So take a minute here to listen to your answers on why you think conformance is necessary and go ahead and put those in the chat pod. Why is conformance necessary? Alright for reliability of a system, to support interoperability standards promote interoperability was the next one too. Exactly yeah conformance is necessary for that. Any others? No? Alright in reducing risk, good. Well conformance helps us build the right system correctly the first time because we are using the systems engineering process and able to validate it using standardized testing. Standardized testing validates the qualified requirements. So one of the chief benefits of using a standard is the interoperability that you get with a variety of vendor products. In order to achieve this you must have a conformant implementation. Therefore conformance is essential for achieving the key benefit again of interoperability that being able to – your systems testing with different vendor products and things like that you get interoperability.

And as once mentioned several times earlier testing reduces risk and that was the next thing that got chatted in there. Risk certainly imposes a potential cost on projects. We usually quantify by the product of the cost of remediation times the probability of the worst case. That would be one way of calculating risk. From a pure project management cost consideration, the cost of testing on the basis of clear requirements can be compared to the risk cost of the device not fulfilling the requirements and having to be rejected potentially after it's been installed. This certainly has happened it's not a theoretical or hypothetical case. The cost of rejecting devices after installation includes the cost of replacements which may or may not be recovered. Might have the cost of legal fees if the rejection results in contractor failure ______ Company will be involved and they'll undoubtedly fight it and that's gonna drive up your costs there. And the cost of not having the benefit of the device for another deployment cycle because you certainly calculated some benefits of having a system in place and if that system gets pushed back for 12 months you lose those benefits. So using standards based testing reduces the overall cost of doing the testing.

Well we're coming up to nearly the end of today's webinar and I just wanna kinda look back at our learning objectives that we had on their first one was to explain to decision makers the need for and benefits of standardized testing. And we went over a whole bunch of those. We have questions coming in so I'm gonna go to the question or the comment before we continue on with the learning objectives. Each step in the testing strategy that I've described makes good sense, that's good. However the aggregate package requires extensive time expertise, special purpose test hardware and software, of course there will be substantial costs. Does it make sense for smaller systems or is this more for mature COTS items?

Yes, the smaller your system certainly the less extensive your testing plan is going to be. But again it's probably a cost well spent. The bigger your system the more costly it is to come up with a very robust testing plan. For a smaller system I still think you need some type of testing plan. Now I'm not saying – even – you could even look at an isolated traffic signal installation, you do have some type of test plan. And in a lot of cases it may not be written, or maybe it was written in the contract that the contractor shall do this and this, and this. I know when I was installing traffic signals yeah it was, we sort of, it probably wasn't documented as well as it should have been but we had them go through a series of tests before we actually accepted the signal after it was constructed. So by default the smaller projects are going to have a less extensive testing plan, but again I think some type of testing plan is probably still necessary. I hope that answers your question.

So hopefully after today you should be able to explain to decision makers how this might kind of tailor into that particular question there. To tell your department head maybe that's a public works director. Maybe it's a city manager, why you need to spend kind of this extra money up front to develop a good robust test plan, because it's going to help reduce risk and hopefully get a better implementation. So by explaining what you've learned today hopefully that will help you in your case for spending a few extra dollars up front in developing your testing plan. Secondly you should be able to describe how ITS standards fit into the overall scope of the systems test in a systems life cycle and we talked about how most of this testing takes place on that ascending side of the V where we're doing our unit testing and our subsystem testing, our system testing, and our system validation. So it can always go back and we do that through traceability.

And a comment comes in on there on that – getting some extra money upfront for testing he won't hold his breath. So good comment, well put it's not always easy. But I think most you know if you can present a very good case you'll have a better chance is I guess what I'm trying to say but yeah in the end most of the time if you present a good argument a cost benefit analysis of why you should spend a little bit more money up front I would hope that they would see the wisdom in that. But you're right it's not always gonna happen and you are given a set amount of budget to make this thing happen. So again the systems life cycle how it fits in there.

Discuss, you should be able to discuss how to test an implementation for conformance standards, conformance to standards, so that's by developing a good test plan. A good test procedure, and having the right equipment and there is equipment available for doing testing and things like that. You're not necessarily reinventing the wheel and there are certain ways you can get that information as well. Finally distinguish the difference between standard conformance, okay that's going to conformance – conformance is to the standard the mandatory and any optional selected standards that you've taken to use to. And then there's compliance with the project, so have they built the thing right? Have we built the right thing? Conformance and compliance. Back to the questions. Are there any examples of good test plans procedures available? Yes. The ITS standards website which I think I have a slide on coming up here has a lot of good information there about testing and test procedures and test plans. Some people I think have even submitted some stuff; it's a fairly extensive website. I would encourage you to check that out for more information on standards and testing and all that. Are there any software for building a test plan? I do not know. I'm not aware of any but if anybody out there, I know there's a few people in the audience that may be able to chat in the answer to that question. So I guess you're saying you're looking for some type of software that helps you build a test plan. You would input some type of criteria and it would spit out what your test plan would look like or at least the basis of the test plan. So again I don't know the answer to that question but for some of you out in the audience there I would think you would know the answer. So yes there's somebody who actually privately said that there is something out there by one of the companies but I'm not gonna promote any particular company's software today.

But I recognize that somebody out there does do it and you can send me an email later if you wanna talk about that. I'll see if anybody else chats anything in as well. As we're nearing the end here feel free to chat in any questions that you may have. The student supplement has some excerpts from some documents that talk about testing as it relates to NTCIP that I've included and that may help you out. There's NTCIP 9001, and again I've just taken out the excerpt of it dealing with testing. NTCIP 9012 also is a good document that talks about test documentation and test execution. In fact that's a very good one to look at and that if you download that I think it's one of the – it's in your student supplement it should be in your materials window, just want to look here, yeah, there is your supplement in there under materials. Well if you want some more information on any of this stuff for standards and testing and systems engineering here are some very excellent websites that you can go to. The www.standards.its.dot.gov that was the first one I was mentioning that has a fairly good database of things in there relating to standards and ITS standards in particular. The ITE website you can go to that one as well ITS architecture and implementation program. That's on the office of operations at the Federal Highway Administration if you want to learn more about NTCIP you can go to the NTCIP website and there's also a very good systems engineering guide for ITS projects on FHWA's website. So again if you are just getting into this and want to learn more there you'll find some great beginner materials, if you're very well experienced and have done a lot of testing there's also opportunities for you to feedback on the RITA website if I remember right where you can implement your, or submit your lessons learned. That one has a very good database on implementing projects related to ITS standards.

Well, that is going to almost bring us to a close as I pointed out we didn't go into any great detail on how to write a test plan. Why? Because that is the next course *T201 How to Write a Test Plan.* And that will get you much more information than you would want, than you would need probably. That will help you at least get going on how to write a test plan. So looking over I don't see any unanswered questions at this point so I will at this point say thank you for your attention today and that this concludes today's webinar.

[End of Audio]