

## **Module 2 - A101**

### **Using ITS Standard: An Overview**

#### **Webinar Transcript**

*Shelley Row:*

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

I am Shelley Row, the director of the ITS Joint Program Office for USDOT, and I want to welcome you to our newly redesigned ITS Standards Training Program of which this module is a part. We are pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this new approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like you.

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 will tell colleagues and customers about the latest ITS Standards and encourage them to take advantage of the archived version of the 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 web site [www.pcb.its.dot.gov](http://www.pcb.its.dot.gov).

Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. Thank you for participating and we hope you find this module helpful.

*Nicola Tavares:*

This training module is A101, Introduction to Acquiring Standards-based ITS Systems. This module is intended for those who perform or who are responsible for the procurement of ITS equipment such as procurement managers, procurement decision-makers, and project managers. Your instructor is Ken Vaughn, president of Trevilon Corporation. He is a leader in the development of the NTCIP standards and has developed several software products to test NTCIP implementation.

The next voice you will hear will be of your instructor.

*Ken Vaughn:*

Once again, my name is Ken Vaughn with the Trevilon Corporation and I'll be your presenter today. Before taking this course, we assume that you have some background. There is a using ITS Standards: An Overview. That is course I101 in the series. That would be useful prerequisite.

It's also helpful to have knowledge of Intelligent Transportation Systems in general, managing ITS deployment projects, government procurement processes, and really just having an understanding of benefits of standards. And a lot of this we'll go over today and systems engineering process as well including the V diagram which will be used throughout this course.

A lot of these concepts are included in that I101 course. So if you are unfamiliar with these, you should familiarize yourself with those topics first. There are two possible paths a user can follow in implementing systems based on ITS standards. The path to be taken is on the standards needs to implement the system.

An identified standard could have been developed using an SEP of well-defined needs, requirements and design, or could have been developed without an SEP. Users should follow the SEP curriculum path to understand how to implement a system if the standard is SEP-based. This is the second module in that path. The first module is I101 Using ITS Standards: An Overview. This module, A101, Introduction to Acquiring Standards-based ITS Systems, and then this module will be followed by A102, Introduction to User Needs Identification. And then you see the remainder of a path there.

If the standard does not include SEP content, then the path is revised a little bit. However, the first four modules are identical to the SEP path. I101 followed by A101 followed by A102 and A201. But if your standard does not include SEP information, it then follows on to A202 and A103 and then on down as you see written on this one.

For this particular course, our learning objectives are to identify what managers should know, particularly process for acquiring standards compliant ITS systems and to differentiate between standards with and without SEP constant. Now, you'll see in the top corner of some slides which learning objective the particular slide applies to, so these are the key learning objectives for this course.

So we'll start out with an activity. Each activity will begin with a slide like this and you can use your chat window to provide your responses. So the first question that we have is what do you think of when someone mentions ITS standards? So go ahead and type in your answer there in the chat window and we will compare those answers. This is really just to make sure that we have a common understanding of the terminology that we're going to be using. So feel free to go ahead and type in your answers.

We're starting to see them come in. The first answer I see is that it's NTCIP. That's one of the ITS standards. There are other ITS standards such as the traffic management data dictionary and others. Standard method to interface with ITS devices is another suggestion and someone

suggested it's a complicated time-consuming process and I guess my hope is that this course will make that process a little bit easier for you.

So the definition that we'll use for this course is basically two different types of ITS standards. One are data standards which include most of the NTCIP—the standards for message signs, weather stations called ESS or environmental sensor stations, traffic management data dictionary, etc. Each of those are divided up by topical areas so they define data, the messages that go back and forth to exchange traffic, or transportation information.

There's also a set of communication standards. These tend to define your low-level communications such as how you communicate over TCP/IP, Ethernet, serial, etc. within an ITS environment. Now, to a large degree, these standards use off-the-shelf broad IT industry standards, but we selected a bigger set of options within each of them so that they're most applicable to the ITS environment and most standardized.

Both have to be defined if you expect your systems to interoperate for any particular interface you are dealing with. Now, there's a couple other key terms we need to discuss really quick to make sure that we're on common ground. The first is management system. When we talk about a management system, we are really talking about a system that is using a service of a remote device and this works very well. These terms work very well for centered field control, for management system uses features off of a remote device.

We're also going to use these same terms rather than switching terms around. We're just using the same terms when we talk about center-to-center communications and that environment now both would actually be acting as a management system and in some cases, and a device. One is providing a service, one is using the service, but rather than switching terminology all the time, we will stick with these two terms.

So that brings us to our next poll and the poll works a little bit different. I want to show you here in a second. The choice is how do ITS standards assist in procurements. There are four options. We'll go ahead and start this poll. So you should now see the poll in front of you with the four options. Do ITS standards define all requirements? Do they define details, but need tailoring? Do communications standards provide all details, but data standards need to be taken?

Or the final option, data standards are precise, but communication standards need to be tailored. So go ahead and submit your polls there. Let's see, most people have responded and everyone—I'll just close up the poll now and show the results. And as you see, everyone has selected the correct option, which is they define the details, but they still need to be tailored.

Well, exactly how do we tailor ITS standards? The standards provide a checklist of features to consider. These include optional features as well as desired ranges, so you can select what range of features you want. So for example, an optional feature may be whether or not I want my message signed to display graphics or not. That's kind of an advanced feature that many signs may not need.

The other case is the desired ranges. Some signs can support many, many message, hopefully in a message library. Other signs may always be connected to essential system or may not need to store a large number of locally defined messages. So once again, there are options there with the standard and when you go there to buy a sign, you need to define exactly what you need for your particular project.

Now, there are many benefits of standards. Not only does it provide you that kind of checklist to go through, but it also provides a lot of management benefits that were defined in Module I101. There are also a lot of acquisition benefits such as price competition among product vendors. Having all of your vendors design their products to meet the same exact interface. There's much more competition. IT becomes much easier to switch from one vendor to another vendor. That increases competition, it should decrease prices.

It also becomes easier to switch from one vendor to another, so over time, one vendor could go out of business if another vendor wins the low bid, it's a lot easier for your system to switch from one vendor to another. It also means reduced integration costs. No longer do you have to worry about getting the protocols from each and every vendor on your system. Now there's one standard that everyone interfaces with that becomes a very easy thing for integration not only for your particular project but for every project across the country.

So you can leverage the benefits of the development of the interface from another system somewhere else in the country is now benefiting your particular project. And then finally, there's market synergies, there's off-the-shelf testing tools and other products that help assist the use and testing of these standardized products.

That brings us to our next activity. So if you go to your chat window again, how do we determine the appropriate tailoring? So we said that this standard is—has to be tailored, but how do we tailor that? How do we determine which selections do we make? How many messages do we need our sign to support? How do we tell we need to support graphics from that? You know, just go ahead and type in your answer to the chat window and then—so any ideas on how to select this? An answer we're getting seems to be based on user needs, which is correct.

User needs—starts out it's really just a V diagram as part of the Systems Engineering Process. This diagram explains that whole process. We're going to be focusing on basically the downward edge left side edge of the V that starts with constant cooperation, system requirements, available design on down.

So, what we're looking at is the constant operations system requirements. You define how you want your project to work. You then start specifying the requirements for your system and then once you start defining your individual components, your subsystems such as your sign or your central system, you then define what features I need out of that sign. That's all related back to your fundamental constant operations. So if you don't have a need to display graphics to your drivers, there's no need for your sign to support that feature.

So, the benefit of the SEP is it helps us to find the scope. We're better able to identify what needs we have to make sure that we specify the right equipment. It also includes a higher level of

stakeholder participation so that we make sure that all of the stakeholders needs are properly addressed as many of these systems are very expensive and we want to make sure we get it right.

Because we're involving these people, it's more likely that we'll hit a system that meets these expectations, but it also means that we'll have a better system documentation \_\_\_\_\_. We'll have complete documentation 'cause we've talked to all of the users. It also means that we reduce the risk of cost and schedule overruns. By properly defining your user needs up front, we make sure we're addressing the real problem, not fake problems. We also make sure that you properly define the problem so that we're properly addressing it. Results are more predictive of outcomes.

So that brings us to our next activity. Who do you think are the key players that are involved in typical systems procurement? Go ahead and use your chat window to define who you think the key stakeholders are in a typical—and really, we're talking about not so much the stakeholder, but the actual players who are the people that are kind of contractually involved in a process like this.

So go ahead and type in your answers and I'm starting to get some responses, the owners and the users. Those are both very good answers. There are, in fact, some others that we'd consider. System owner is a very important one and that really includes both the actual owner as well as the users or the system operators \_\_\_\_\_.

The second key player that's very important is the systems engineering assistant. The systems and engineering assistant may, in fact, be part of the system owner's staff or it may be an external consultant. The key is that they have the understanding of the kind of IT side of the environment rather than the traffic engineering side of environment. They understand the process that you need to go through to have a successful procurement for an IT system and they should have knowledge of ITS, but the real focus is in the process to acquire a software system.

And then finally, you also need a development team that will actually code and implement the system. Once again, this may be part of the owner's staff, but very typically, it's an outside firm either for a sign vendor or a device vendor or for a \_\_\_\_\_ system developer.

So interaction among the team, communication among these three players is critical. Many projects have gone awry when one player has either been left out or for whatever reason, has not been able to make contact with the other players or their role has been minimized. All three roles have very distinct perspectives and skills and they need to be considered. When they raise issues, those issues need to be properly addressed and taken into consideration.

A lot of times what may seem like a small issue to one person may in fact turn out to be a very big issue. So, every issue raised needs to be properly addressed.

So it's important to identify those roles early on in the project rather than waiting till the end. Once they're at the end, it becomes very difficult to get them addressed in a cost efficient manner.

That brings us to our next poll. Where do the standards fit into the SEP V diagram? So I will watch that poll. So go ahead and make your selections there and there are four more choices given either at the top of the V diagram, concept of operations, system requirements, and high level design. High level design and detailed design or ITS standards address issues outside of the V diagram.

So go ahead and make your choices and I will close out the poll here. And you see that we have—I'll share the results now. You see that we have somewhat of a disagreement. A couple people said, or two thirds of the people said, at the top of the V diagram and the third said in the high level design and detailed design. The reality is while we are talking about user needs and system requirements, we're really talking about the subsystem level, but we're talking about standards. We're talking about an individual device or central system.

So we've already defined that there is a larger system. We're now dealing with a subsystem definition. So, if a subsystem definition, we are really talking about the high level design and detail design. And if you really think about this, this really starts making sense, 'cause after all, the detailed design is where you start defining the how component.

So if you look at the V diagram, I'll hide poll results here. If you look at the V diagram, you actually see the entire focus of ITS standards should be to define how a system works and the detailed design is going to define the bits and bytes that go across the wire that is a very detailed design. That's the actual intent of the standards. For those standards that contain user needs and other systems engineering content. That relates to the high level design and subsystem requirements 'cause they're really only focused on the subsystem and then you look at the other side to it, the other \_\_\_\_\_ subsystem.

Now, once again, some standards also include test procedures now. They would be shown here in the diagram as part of your subsystem integration and subsystem verification steps of the \_\_\_\_\_. So those are the areas of the V diagram covered by the standards and if you notice, I purposely have drawn them only on the edges to reflect the fact that it does not define the entire set of system requirements or detail design. It's only defining one small piece of that step. I think there's a lot more work to be done. Standards have done part of the work for you.

So for those standards with SEP content, they define subsystem user needs such as managing the fonts for a message sign. They also define subsystem requirements. That's more detailed analysis. It determined the number of fonts, determines the number of graphics, makes sure how large each font set is, etc. And each one of those requirements, the standard will also trace back to the defined user need. And then finally, it defines a precise requirement and it traces each requirement to a single design. So \_\_\_\_\_.

Now, standards without SEP content, and these are mainly your earlier ITS standards, the content was derived by perceived needs. Unfortunately, they were never actually formally documented as a part of the standard. As a result, the context has to be inferred by the user. This can largely be done by looking back at the performance groups in those standards and saying, “Okay, which design elements are grouped together in a performance group?” Now I can figure

out why this performance group exists, what purpose it fills, that really essentially is the user need it's related to.

So those components need to be defined by the actual user in specifying these standards. And you define the user needs, requirements, and potentially even in some design, details such as the precise messaging sequence goes back and forth. So that's quite a bit of additional work by the user community and it may be that your staff doesn't have that expertise on hand, in which case you may need to go rely upon outside sources to find that expertise.

That brings us to our next poll. How rigid are subsystem requirements? So you should now be able to see that poll. How rigid are subsystem requirements? What type of contract do you use to require the subsystem? Do you assume that your requirements are known? Do you introduce a fixed price contract or requirements will be revised so we should use cost plus contract? The third option is it depends. So go ahead and close that poll and share the results.

Most of you selected it depends, which is the correct answer and we'll discuss that in a second. A few of you suggested requirements are known or requirements will be revised and the answer is, in some cases, they're known. In some cases they need to be revised so the actual answer is it depends.

Now, the reason is because one of the things it really depends on is whether we're talking about a well proven device or if you're talking about a prototype deployment or something like a central management system which tends to be very customized for each project. The more customized something is, the more likely it will need to refine the requirements as you move along the project. And in that case, you'll need cost plus contract rather than a fixed contract.

If you know your requirements aren't going to change, you can fixed-priced contract. But devices are largely off the shelf. Requirements are well known for those devices, so you can use a fixed price contract. Management systems often require development and therefore, you'll need a cost plus contract. So different scope requires different \_\_\_\_\_ and a different way that we interface with the V diagram Systems Engineering Process.

So, for example, the typical scope by spender is the agency along systems engineering system defines your cost of operations, defines all of the system requirements for the entire end-to-end system. They then define the various components such as your message signs, such as your central system, sensor stations, etc. And for each one of those subsystems then defines requirements.

So if it's a device, something like a sign, you can find detailed requirements for the sign. That project then goes to contract and your device vendor has a very well defined set of requirements that they need to fulfill. That is performed by the device vendor largely behind the scenes. They go off and use the device and all of a sudden, they return with a device that should be largely conformant to your—or compliant with your project specification.

We then go through subsystem verification to make sure that device works properly and at that point, your vendor is working with the client both in a very open environment, a transparent

environment. You can see what's going on to identify any problems. Once it passes subsystem verification, typically speaking, your device manufacturer gets paid and his job is done.

Now when you compare that to a systems integrator, it works out very different. The agency starts to define a concept of operations, starts to define system requirements, may even start defining high level design and subsystem requirements 'cause after all, we're talking about the details of an entire system is to start defining exactly what the central system's supposed to do as opposed to the system overall.

But at some point, we bring in a systems integrator. We hire that systems integrator. That systems integrator, more than likely, is going to bring in his off-the-shelf software and that off-the-shelf software will be designed using certain assumptions. So what we'll see is that that systems developer will want input now into the concept of operations, system requirements, high level design on down. At those levels, the interactions tend to be very transparent.

Obviously, the system's integrator is just changes the concept of operations. That needs to be made very aware to the entire project team. But as you move down, as you get into the detailed design, more and more of this starts becoming more by the systems developer separate from the project \_\_\_\_\_.

So two very—well, I'm actually going to go back to that. Two very different designs. As you work back up, you'll also notice that the systems integrator is involved in the project for a much longer stage as well. He will be there through subsystem verification, through systems integration. This integrator is responsible primarily for that integration verifying that the system works as specified and even on board to make sure that the end users are getting a system that meets their needs. That's called system validation.

So if they find something that does not meet their needs, that system vendor still needs to be on board so they can go back and fix it. And that's undesirable and you don't want to have those cost \_\_\_\_\_, but the system vendor still needs to be there to take care of any issues that arise at the last moment.

So ITS standards really the fundamental concept behind the standards is to reduce the work, simplify project specification. Yes, it's complicated, but it's much more complicated if you don't have the standards since the starting point. It allows reuse of the design and implementation because the same standards will be used all across the country from one project to the next. That means you start getting much more stable software.

It also means you have facilitated testing because the same test procedures can be used over and over again and more than likely, your software has been tested on other projects, which means your first test has a higher chance of being successful or mostly successful rather than having to work out bugs every single time you turn around. Because this reduces work, it's also reduced risk, it reduces your schedule.

So how does the procurement play out? Well, it kind of depends if you're procuring a device or a management system because of where, how stable those requirements are and when the vendor is

hired. It also depends whether your standard you're dealing with includes SEP content or it doesn't. So there are four distinct scenarios there. We'll go through those.

All four of the scenarios start out with certain preparatory steps. These include defining the concept of operations for your project. That's a very high level concept. So it's just informing the public about current traffic related events. We then move on to system requirements. Well, let me talk—one thing about the concept of operations is that it is very conversational. It is intended to speak and use the terminology of your system users and stakeholders. It is a very readable type document.

We then move into system requirements. System requirements are very precise definitions that are really designed to be written with the tester as the audience. Anything that's written in a system requirement needs to be measurable and testable so that at the end of the project, you can properly sign off that it met that requirement.

So here, you often see the words like shall. The system shall allow the user to define the message to convey to the public which will automatically expire when it—when the event ends. So it's a very precise definition to be measured. We then move on and start defining your major subsystems. So up to now, all we've said is that the system as a whole needs to get information out to the public.

Now we'll start saying, “Oh, well, we will achieve this through a management system and through message signs in the field.” Could it be done through \_\_\_\_\_ displays or other things? Our particular design has said we will have a message sign subsystem and that's the way we convey that message to the user.

We then also need to be concerned about defining the communications environment and this includes whether it's a TCP/IP Ethernet type environment or whether we're using serial \_\_\_\_\_ on all of those details. Finally, we identify the services needed from the external subsystems.

So if we want to display the message to the user via message sign, what features of that message sign do we need to be involved? And that's where we start diverging paths. If you're procuring a device and the standard includes SEP content, then the selection of those features is fairly straightforward. You can go into the standard. You can identify the user needs out of that standard such as define a message and activate and display a message. Those are two features defining the standard, user needs.

You then—those user needs will then trace to specific requirements within that standard such as support multi-page messages. So there are maybe 15-20 different requirements that are all related to defining a message. Which ones do you need? Which characteristics of message definition do you need? Do you need multi-page messages? Do you need multiple fonts in the message, multiple lines, multiple colors, etc.? So you go through and the standards provide all of these options as—in a single table.

You can go down the list and identify which options you want. You map the design elements already done by the standard. So once you say I want a multipage message, there is a defined standardized mechanism, single design that the standard provides for that solution.

Now, if your standard does not have SEP content, it's a little bit more difficult. You need to define those user needs and services you need \_\_\_\_\_. You then need to define your requirements; map them back to your user needs. Finally, you need to map your requirements to the standardized design elements, recognizing that some of your elements may, in fact, be missing, may have some feature that you absolutely require and the standard's either ambiguous or has a missing feature. So you need to fix that aspect as well.

So in particular, some of the older standards did not define the exact messaging roles and how you package elements together to achieve the goal. So that's extra information that you need to define if you're using a standard without SEP content. Once again, if you don't feel like you have the expertise to do that, then you may need to consider going to an outside source to provide those details for you.

The third scenario is standard with a management system using a standard with SEP content. This is fairly straightforward process as well. You select user needs from the standard, select your requirements from the standard, and then define the scenarios when the exchange is required.

So that's an extra step from the device. Not only do you have to define user needs and requirements and automatically selects the design details, now you have the added step of saying under what scenarios do I use this feature. So, for example, if I want to monitor a message that's on the sign, that's a feature that's defined within the standards, but is that an automatic background operation that you want your system to do always to ensure that the right message is displayed on the sign or is that a manual process? The standard is silent on those details. That is a management level feature that is specific to your type of project.

So, lots of details there. When do we know when we should document the need for standardized features? As you may remember, at one particular point, we defined you're going to hire a systems integrator to come on board. Ideally, you want to give that contractor as much knowledge as possible as to the type of system you want to procure. So you want to provide them as much details as possible so that they know what sort of system they need to provide and that will affect their bid. However, if you forget something, remember that there's a whole process to define the information.

The key is anywhere along this vertical line, you have that opportunity to harmonize your documents so even if you get down to the design details, you still have the opportunity to discover, oh, there's this detailed feature. I need my signs to support graphics. I hadn't thought about that in my concept of operations. Now, let me go back and view my concept of operations so I can better explain why I need the feature of graphics so that I have just \_\_\_\_\_ including the graphic feature like \_\_\_\_\_ requirements so that the \_\_\_\_\_ detail design.

So it's very important that you constantly harmonize all system documentation so that once we get to the end of the project and the testing, if something fails, we know the impact that has on the project. If a graphic feature fails, I can then trace back and say, "Well, I know the requirement failed, but how does that relate back to my concept of operations? I've traced all of my user needs all the way down to my design details. When I go into testing, if something fails, I can immediately identify the practical impact that that will have on my concept of operations on a day-to-day basis.

And throughout that process, it's the harmonization and coordination with your vendor that makes all that work.

So the next concept is the standard without the SEP content. If you're a management system and you're not using SEP content, then once again, this would be similar to your device without SEP content. You'll have extra work to do. You need to define your detailed requirements, define your detailed user needs, map the exchanges to design details and enhance the dialogues, the messages, the data elements as necessary.

Once again, this requires a fairly detailed understanding of the standards, how they work, so if you don't have staff on hand, you need to go out and find the resources to do that for your project. And then as with the other management system, you still need to define the scenarios when the data exchange \_\_\_\_\_.

Now, in the real world, a management system is likely to handle devices some of which are reliant upon standards that contain SEP content and others will have standards that do not have SEP content. And in fact, you may have some standards for which interface standards don't even exist.

So you have to deal with all sorts of devices. You have to address each one of those issues separately. But all projects, every single project, should always follow the SEP process, Systems Engineering Process. And use that content to document that information within your project where that content is already worked out for you in a standard, you're ahead of the game. You can just reuse that information. But if it's not already defined for you, you need to take the time to define it for your particular project to make sure that you will get a system that meets your needs.

So once you have all that done there are still on all four scenarios these following steps. You need to then define, after you define all of your user needs, all of your informational requirements regardless of the scenario, you need to define your communications stack and standards related to this communication stacks. You need to define other requirements such as your hardware environmental requirements. You need to actually go through the procurement process, implement the system, and then finally test the system.

Now, if you're testing standards with SEP content, sometimes you'll discover the newer standards have or updated standards have test procedures in the standards themselves. This is already true for the environmental sensor standard and it soon will be true for a message sign standard. That facilitates testing, know the exact test procedure to follow. It means your vendor is likely to have

been tested to that process already and it also means that it's going to be much cheaper to test 'cause there will be standardized tools in the marketplace to assist you in testing this product.

There are other standards that even though the test procedures aren't included in the standards themselves, there are still standards available or test procedures available in the industry. They've been deployed in other locations. People have already written test procedures and by and large, those test procedures likely could be used on your project as long as the standard follows SEP content. Why? 'Cause those standards or those test procedures that have been written to test the requirements written of the standard.

As long as those requirements are written in the standard as part of SEP content, that same test procedure should be able to apply to your project. So it can be used repeatedly and likewise, it's very likely that there are tools available already in the marketplace to help you do that sort of testing. But for those standards that do not contain SEP content, that means your requirements are going to be a little bit more ambiguous and therefore, the actual test procedures written for other projects may or may not apply to your particular project. They may have to be tweaked a little bit. Nonetheless, it's worth looking around the industry to find out what's available and then to reuse as much as possible.

So testing the final product. Testing cannot be overemphasized enough. It is a critical step. Too many projects go forward, you know, you're at the end of the project, there's pressure to move the project forward. You need to make sure what you're getting actually meets the requirements that you specified. Verify the subsystems meet standardized the interface. Then verify the system integrates all the components together. Then, finally, you can validate that the entire system meets the user needs and works as was originally envisioned.

You also need to make sure you document all the testing. You need to document your test plan before you start testing. You need to document all the results as you're testing. That allows you to reproduce the same results over and over again. If you have a defined test plan, defined procedures, you know what you're doing every step of the way and it's much easier to reproduce that process by defining it up front.

Likewise, by capturing all of your data as you go along, you can identify exactly when something went off track and identify any ambiguities or any anomalies. Also, always remember, you test in order to find problems. Systems are extremely complex. You very likely will find problems when you go to test. So you need a schedule or multiple rounds of testing both schedule-wise and budget-wise.

This allows you to identify problems during your testing and then have the flexibility to wait for another round of testing before you deploy the field. You can only schedule a test—you need a schedule to do anything else, the assumption is it's going to pass. And the real purpose of testing is to find out what doesn't pass.

So that brings us to our next activity. What are the practical impacts? And you can use the chat pod to answer. What are your concerns about applying the systems engineering process as we've described to acquire standards-based systems? So from a practical perspective, we've described

the process from a practical perspective. What are your concerns at this point? So please feel free to use the chat pod.

So far, I'm not getting any comments back so I'll wait a few more seconds and move on. Or you can type in your questions at any time during the process, remember. So I got one concern, which was locating relevant standards. If you're talking about devices, you can go to the [www.ntcip.org](http://www.ntcip.org) website and go to their library and all of the standards are listed out there. The other major standards are [Traffic Management Data Dictionary](#) and instant management data dictionary. ITE is an excellent resource to go to. [ITE.org website](#) is an excellent resource to go to find out that information and there is also the [ITS standards website](#) by the Federal Highway Administration. All of those resources are in your student supplement.

Some of the other concerns we've heard in the past include how large is the resulting specification. And the answer to this really should be specifications need to be as detailed as necessary. If you think a feature is important to have in your system, then it needs to be identified in your concept of operations. You also need to define it in your requirements and trace it. Then, what's the purpose to define a requirement if you're not going to test for it?

So you need to verify all of your requirements in a test procedure, trace that test procedure back to your requirements. Finally, you need to make sure that it meets the end user needs, so you need to validate at the end of a project so that the actual users identify that yes, the way this works is acceptable to how I want. And finally, you must recognize that this is a process, it does take time and it does take money. So you need to budget and schedule for that effort.

Now, what are the cost implications? SEP does require time. It also requires experienced personnel. You need to make sure that the staff, your IT staff, understands the process and is willing to undertake that task. Finally, it requires commitment. There have been many projects that have started out saying, "Yes, we're going to use the SEP," and then as you get into the process, they start feeling like they're running behind schedule, so they decide to short circuit the cycle, they cut off testing, they cut off some documentation work, and when the end product is delivered, all of a sudden they realize they don't have the produce that they envisioned.

It is well worth the effort once you start down this path to make sure you're following the systems engineering process and stay committed to that process all the way through.

It is proven to lower the risk and increase the quality of the project. Just finishing your project on time is not worth very much if the final product doesn't meet user needs. So lower the risk and increase the quality. Yes, it costs a little bit of an investment, but it's very worthwhile investment. To give you an idea of that investment, this chart demonstrates that those projects that had spent a considerable effort on systems engineering, say in the range of eight to twelve percent, show a very serious significant decrease in the amount of budget that is actually expended on the project. Don't have any systems engineering in the bulk of a project, you can run easily two times to a project of estimate costs because you end up building something that's not the right thing, you have to go back and fix everything.

By spending about ten percent of your budget on systems engineering or maybe twelve percent of your budget on systems engineering, you can bring down the excess cost and use your dollars wisely.

So, Systems Engineering Process reduces risks. This is true even when you're acquiring a DMS 'cause a DMS project itself requires risks. You start specifying features you may not need. You can easily forget features that, in fact, you do need for your operations. So by going through this process, you make sure you specify the right features and you get them without fighting too many features. Now, risks are higher for standards without SEP content. That's because a lot of that work's not done for you and it's not standardized. So if you just go out and buy a generic sort of device, you're much more likely to have a wide range of devices out there 'because they have not been standardized yet.

So it's more important to follow SEP for those devices. On the other hand, it's going to be more effort as well, so SEP should always be followed. Risks are higher for custom development, so for central management systems, for example. So anytime you're working on those systems, you need to budget a little bit more money to look at your systems engineering process to make sure it's still right. And then finally, risks are higher when dealing with multiple standards. Once again, management systems, because you're going through this process multiple times, you really need someone experienced to understand and make sure you're hitting all of the right points.

So today's objectives were to identify the key concepts a manager should know, describe the process for acquiring standards compliant ITS systems, and we differentiated between standards with and about this Systems Engineering Process.

So what did we learn today? Well, first, all the projects should follow—hopefully you all get this, the Systems Engineering Process. The Systems Engineering Process assists in defining the scope for a project and meeting the project budget and schedule.

There's already a hint provided for you on this one. ITS standards with SEP content reduce systems engineering effort on a project. ITS standards without SEP content still assist in projects using the SEP. And all requirements should be fully tested prior to acceptance. So those are the key points from our presentation. You can also learn more from the NTCIP guide, the TMDD Guide, IEEE 1512 Guide, and the Systems Engineering Guidebook for ITS.

Other ITS courses include A102, which is the next module for all standard curriculum paths and A201, which also follows the A102 project for all paths. You may also be interested in testing which is defined in T101. With that, I will open up for questions. Feel free to provide any questions you may have using the chat window and we'll stay on as long as we receive some questions.

One of the questions that I frequently hear is what sort of time commitment is needed for the separate and the answer is it largely depends. As we discussed, it depends on the type of device you're procuring, whether it has SEP content or not, whether you're buying a central system versus a device and how large the project is overall. That chart we looked at provided a range for

kind of the larger systems. If you're looking at maybe a messaging, you probably don't need to spend ten percent of your budget on just something generic.

You'd probably only spend a couple percent on systems engineering, but you still need to look at material. Good news is – a lot of the work's been done for you. But you still need to make sure you're following the process. So go ahead and type in your questions and we will answer those as they come in.

Just looking back through my notes, another type of question we've had is how do I really find out how to specify it, how a sample test—a sample procurement specification would be? And a lot of those details are defined in the following modules. They get into much more detail about how to specify the specific devices indulging all the way down to the very detailed options of the supply system. And with those standards that have SEP content, that's a very detailed definition. For those standards that do not have SEP content, that information requires a lot more focus \_\_\_\_\_.

So, I do not see any other questions coming in, so with that—well, here's one question. Is it fair to assume that all NTCIP standards are standards with SEP content? Can you provide an example of a standard without SEP content?

Well, this changes over time. But there are currently some NTCIP standards that do not have SEP content in them as far as final group standards. I believe the signal control standard still does not have full SEP content. There are probably some others as well, but that's information that changes with time, so the best thing to do is to go to the [ntcip.org website](http://ntcip.org), download the current version of the standard, and identify whether that's good or not. I believe the supplement that we passed out does include a current list of those standards that do have SEP content.

Another question I received, what do you think about open source's approach? I assume this is in relation to software development and in particular, I would assume this is related to management systems. I'm unaware of any projects to do open source or devices at this point, but for management systems, there are some efforts to do some open sources.

As long as you think those efforts will meet your needs, then they're worth considering. Recognize some of the projects that are comparatively open source are—actually, if you start reading the details—are not quite as open as a lot of vendors would like them to be. They are really pretty well tied to one particular system's integrator with somewhat open licensing rules, but you're really kind of stuck with one particular systems integrator.

To me, any true open source approach needs to be available for any systems integrator to some into to develop and enhance things without requiring special licensing fees or other \_\_\_\_\_. So as long as the approach follows that scenario, it's worth consideration. But I think the open source solutions out there right now may have some ways to go before they're really competitive on the full range of features as other mainstream solutions. But that will likely change over time, so you have to look at your particular project, I think.

And with that, that's all your questions I have received, so thank you for attending this course and that concludes the A101 Module.

*[End of Audio]*