SAMANTHA SAULD: So thanks for joining this webinar entitled Practical First Steps to Achieving Web Accessibility and Reducing Liability Risk. I'm Samantha Sauld from 3Play Media, and I'll be moderating today. I'm joined today by Jeffrey Singleton, a principal at Converge Accessibility. And with that, I'll hand it off to Jeffrey, who has a wonderful presentation prepared for you all.

JEFFREY SINGLETON: Hello. My name is Jeff Singleton, as Samantha mentioned. I'm the co-founder of Converge Accessibility. And I've been working in the IT field now for well over-- sorry about that. Somehow my slides jumped. For well over 25 years, I've been working in this field. And for the last-- I should say along with that, I've primarily been focused on accessibility for more than half that time. So I've had some time to really kind of cut my teeth on the accessibility aspect of things. But we'd also like to start off by thanking 3Play Media for the opportunity to present today's webinar as well.

Now, we think you're going to find this information that's being presented today beneficial, especially if you've been struggling in establishing an accessibility baseline of your own web presence, or maybe you've been worried about being the target of an accessibility legal complaint. Or it could be that you've just been struggling with WCAG and how to incorporate accessibility into your web content development life cycle. And the fact that you're attending today's webinar kind of tells me that you fall into one or more of those categories.

So our goal today is to introduce some simple things that you can do to start working toward web accessibility, and at the same time, reduce the legal risk of-- or the risk, rather, of legal complaints by these serial plaintiffs that are quite common out in the world today. And we also want to discuss how to go about incorporating a culture of accessibility into your web life or website content lifecycle in order to help ensure that your content is made accessible from the start, but that it can also remain that way as updates are made.

Now, a Chinese proverb states that "a journey of 1,000 miles begins with a single step." Well, in other words, to reach a goal, you must begin the journey. But where do we start? Over the years, I've seen thousands of people attend various accessibility conferences looking for that specific guidance. Yet, most of the presentations that are given cover very specific topics, and rarely do they ever give you enough information to actually go and then just start the accessibility journey.

Instead, those who are new to accessibility often end up more confused than they were before they attended the accessibility conference in the first place. And just understanding WCAG alone, that can be like pushing a large boulder up a hill, just like what we have illustrated on the slide of our stick figure big businessman pushing this large rock up a hill. Maybe you feel that way when it comes to some of these WCAG guidelines that we have to deal with on a regular basis.

Now, there are three things we want to share with you today. First is an accessibility statement. And we're going to cover why such a statement is important and what you should include in your statement. Then we're going to outline a simple approach to begin establishing an accessibility baseline of your current online presence. Lastly, we're going to discuss a method we have used to help some very large organizations incorporate accessibility into their design, development, and quality assurance teams, or rather, to be able to include accessibility as part of their organization's culture.

Well, let's discuss the accessibility statement first. An accessibility statement is likely the most important first step you can take regarding the accessibility of your site. A simple accessibility statement can help avoid litigation, which can also save you time and resources as you work toward making your site accessible, rather than spending those resources defending against the legal complaint, then just quick-fixing issues to appease a specific plaintiff. Now, unfortunately, an accessibility statement is something that most organizations fail to consider, especially in the very beginning.

An effective accessibility statement really includes three parts at a minimum. First, there's the accessibility policy. To have an accessibility policy means that you're going to be bringing together people from your legal team, your customer service team, from your IT department and so forth. You get all the players involved. And it forces these groups to have a conversation about accessibility, and it forces everybody to agree on a common goal. And this really kind of helps lay the groundwork for an organization's accessibility efforts.

Now, the accessibility policy, it includes two components, an accessibility commitment and an accessibility plan. Now, your accessibility commitment sets forth the organization's commitment towards accessibility, obviously. And for instance, that may include a recognition that your organization cannot discriminate against people with disabilities, and it may identify the various laws or regulations that set forth those requirements. And in so doing this, it publicly demonstrates that you, as an organization, are aware of your responsibility.

Now, your accessibility policy should also include your accessibility plan, and that plan lays out your goals and your milestones for achieving accessibility. For instance, your plan may identify that your goal is to achieve compliance with WCAG 2.1 level A & AA for all new content and major customer use cases by a specific date.

And just be careful here because you never want to state that your site is fully compliant, because this is rarely ever going to be true. Because even if you get to that point with ongoing updates and adding content, it's quickly going to have some issues appear at some point. So also, in stating that your site is compliant at any time, that kind of sets forth the challenge to some out there to prove you wrong. And that's likely to be brought to your attention through a legal complaint rather than a friendly email.

Now, second, we want to provide contact information and a what we call a backdoor for accessing the goods and services that your website makes available. And this is really absolutely critical. In the United States, the Americans with Disabilities Act and other disability rights laws and regulations require something called effective communication, and this includes your website. And just about every civil rights law around the world requires equal access to your goods and services.

So providing an email address and a telephone number, together with your hours of operations, is essential. And providing this backdoor is also a very easy non-technical way of achieving this effective communication in your organization for your web content. It also helps ensure that those with a need for accommodations can still gain access to your organization while you're working on making your website accessible. But please note that this backdoor we're talking about, it doesn't make your website compliant with the accessibility guidelines such as WCAG. It's there to ensure that your users can still access your goods and services regardless of what accessibility issues may currently exist on your site.

Now, third, as a best practice, we want to provide some type of feedback mechanism through which your customers can give you feedback about your site. And this can help to identify content that you may have overlooked in your website accessibility plans. And having a clear complaint resolution process through this feedback feature can informally resolve a lot of issues before they ever end up in a courtroom or in front of some type of enforcement agency. Plus, using a feedback mechanism is a great way to get positive feedback as well, and that helps to justify the money that you're spending on web accessibility.

Now, the W3C provides an online resource that includes an accessibility statement generator tool. And that URL is www.w3.org/WAI/planning/statements. Now, this is a great resource if you need to create an accessibility statement. But that generator tool does get a bit detailed. And in my opinion, you do not need to initially go to that level of detail.

So don't let this get you stuck into an accessibility statement black hole. Instead, stick with the minimum, just as we've outlined, until you're further down that path of having an accessible web presence. Because at that point, then it's going to be easier, and it's going to make a lot more sense to include all the other additional details that this generator tool will present to you. So start out by keeping it simple.

Now, all too often, organizations turn to an automated accessibility scanner as their initial approach in identifying the accessibility issues on their site. Well, it's important to note that these scanners can only evaluate a small percentage of what is covered by the WCAG criteria. And they can often lead to spending a lot of unnecessary time trying to fix issues that are not necessarily critical, and can also provide a false sense of accessibility compliance.

Now, why do I say a false sense of accessibility compliance? Because many blocking issues cannot be identified by these scanners, and they often go undetected when using only a scanning tool. And we have seen too many cases where a website owner ends up being sued, even though the scanners that he's using indicates that there's no issues. And additionally, these automated scanners often report issues that are false, and this can lead to a wasted effort, especially for those who are new to accessibility.

Now, that's not to say that accessibility scanners do not have their place, but we'll cover a little bit more on that in just a moment. When first determining the accessibility health of your website, manual testing is another effort that really is critical, way more so than accessibility scanners. And that should be your focus prior to investing in some type of expensive scanning tool.

Now, in order to understand where your current website sits from an accessibility perspective, you must test it manually in order to establish an accessibility baseline. There's just no way around that. But how do we go about effectively and affordably doing that, especially if our site contains hundreds or even thousands of pages? Well, we have to create a plan.

The first thing you should do is ask yourself, well, why do people go to our website? And then, in answering that question, we're able to quickly start to identify the features of our site that really matter most to our visitors. And next, we want to account for our most visited web pages. This information is easily obtained through our web analytics. And it's also important because if most of our visitors are ending up on those pages, we want to make sure that those pages are accessible.

Now, most websites today do run on what we call a CMS, or content management system, and this means that the site is made up of a series of templates or layouts. And ensuring that we are including representative examples of those templates and layouts that are being used, in addition to the various types of elements, or content, or even states that those pages can have, is going to make sure we're identifying issues that show up throughout the site. And this is going to allow us to address issues that can be fixed in one place and then cascade throughout many pages. So we kind of look at that as the low-hanging fruit. It's really the framework of our site. And identifying those and fixing them can really give a huge boost to the accessibility of our site.

And we also have to think about the critical user scenarios. And what do I mean by that? Well, for example, if we have an e-commerce site, a critical user scenario might be this. A user comes to the site, searches for a product, finds that product, and reviews the details of that product. And then he adds that particular product to the shopping cart and proceeds to go through the checkout funnel to purchase that product. Now, this scenario in this case needs to be accessible from start to finish, regardless of what else might exist on the page, because that's a primary reason that the user came to our site in the first place.

Now, an example of what we might consider a secondary user scenario-- we'll use our e-commerce site as the same basis for our example-- would be for signing up for a newsletter. Now, obviously, that's not the primary purpose of an e-commerce site. But while signing up for a newsletter is something we still want to make sure is an accessible experience, it's not as initially critical as that first scenario.

And so just remember that our goal initially is to whittle down the required testing effort to a manageable size. And in this way, we can more effectively establish an accessibility baseline and reduce the feeling of overwhelm that often comes when first starting down that path to accessible content. And you may have experienced from using a scanning tool the amount of data you can get. Well, we want to reduce that data down to something that's manageable as well. And that's where this manual testing effort comes in.

So using this method to establish a testing plan for our initial accessibility testing allows even complex sites, sites that have thousands upon thousands of pages, to be reduced to a manageable effort. In most cases, what we've seen using this method to reduce a large site down is that we end up anywhere from 12 to 30 pages and maybe three or four critical user scenarios that really require us to do that manual testing. And this makes our initial efforts into establishing our current state of accessibility much more affordable and much more attainable.

Now, once we know what pages and features to review for accessibility, what are we going to do? Well, we have to do that manual testing. And we don't have time to cover the specifics of manual accessibility testing, but we can give you a few tips. Now, if you're not sure how to perform manual accessibility testing, then you really should consider working with an outside accessibility services company, such as Converge Accessibility, to help with that initial testing effort. And this can often help jump-start your efforts, because if you work with the right consultant, they can also provide an education on this topic for you and your staff to really help you get a good grasp and baseline of the type of issues you have, and why they're issues, and what you can do about them.

Now, when it comes to manual testing, as we mentioned, it's manual testing. But unlike a lot of manual testing, accessibility testing often requires multiple test passes. So just be aware of that. And one of the first and foremost things we need to look at when it comes to accessibility testing is doing that keyboard testing. And that's another area that often gets overlooked when it comes to accessibility.

Now, anyone can do this type of testing if they're given a basic understanding of what they need to be looking for regarding accessible keyboard content or keyboard accessible content. And it's really quite simple. Using only the keyboard, you want to verify that you can always visually track the element with a keyboard focus and that any interactive controls can be manipulated successfully using only the keyboard. Also, many assistive technologies rely on content to be keyboard accessible for them to work properly.

Now, if you go to your current website and you use just your Tab key to navigate through that site, and then you lose track of what controls have focus, or if you try to interact with one of the controls to accomplish a task and you're unable to do so successfully with the keyboard, then you have a serious accessibility issue, one that would be considered a blocker in most cases.

A screen reader testing is also a must. And so far, we've covered topics that are pretty simple in practice. But unfortunately, screen reader testing comes with a bigger learning curve in order to perform that testing properly. So if you've not done screen reader testing previously, again, I encourage you to work with a third-party accessibility services provider, such as Converge Accessibility, in order to get a better understanding of what that type of testing entails, what kind of issues you can discover doing that type of testing, and what kind of recommendations you have to address those issues. So we encourage you to choose your third-party consultants wisely, because unfortunately, there's many out there that will just throw a report at you and then walk away and not provide any further guidance.

Now, there are some optional things you can do initially, and they are good to do at some point. And one of those is doing screen magnifier testing, along with voice command testing. But again, this is not something you need to worry about out of the gate, but it is something to consider as your manual testing process matures and gets further along.

Well, what about automated testing tools? Well, depending on your level of knowledge around HTML, CSS, and the WCAG guidelines, this can also be a fairly simple exercise, but not always. So keep in mind that these automated tools should not be used as the primary means of testing or validating the accessibility of a site. They really are just tools. They're not the end-all be-all, regardless of what the tool vendor salesman might tell you.

Automated testing is admittedly limited as well. And there's only a handful of the accessibility criteria that can be thoroughly tested using automated technology. Yet, it just so happens that this small handful includes some of the most important accessibility requirements to your legal risk exposure. And why do we make that claim?

Well, some years ago, my colleague and the other co-founder of Converge Accessibility, Ken Nakata, researched all the legal complaints and settlement agreements that he could find on web accessibility. And then he asked the question, what is it that got these organizations in trouble? And he found that there were four issues that kept popping up over and over again in these complaints. Now, these four problems are inaccessible images, and inaccessible image maps, bad table markup, and missing form labels.

Now, while our list includes alt text for image maps, we have not seen that much recently. And that's likely because image maps are not as widely used as they used to be. Now, if I had to venture a guess as to what would replace that particular item on this list for the complaints that we're seeing today, I would have to say it would be the inappropriate use or the lack of use of headers in navigation regions. But fortunately, this is also something that can be easily identified using an automated scanner.

Now, it's impractical to manually test every page on a website if you have thousands or potentially millions of web pages. But a good quality automated testing tool can test that kind of amount of content for you fairly easily. And because these four issues that we just mentioned can all be tested via scanning solution, it makes sense that automated testing should be part of any serious accessibility testing effort. Another point to consider is that drive by plaintiffs are also using automated scanning tools to qualify a website as a potential target for their legal complaints.

Now, we know that this really does go against the end user license agreement for most of these scanners, but that doesn't stop them from doing it anyways. And using automated testing tools effectively, it's going to help keep you off the radar of these professional plaintiffs if you're addressing these common issues that are found by such tools. Because if these professional plaintiffs aren't finding those, they're less likely to look at you as a target. But again, an automated scanner is not something you should probably rush out and invest in immediately, unless you really have a high concern about receiving some type of legal complaint.

Now, performed periodically, automated and manual testing combine to create a regular rhythm that improves accessibility. But it doesn't infuse accessibility into the organization's culture. And that's because these approaches alone are purely reactive. Instead, what we're trying to drive home here is that creating a culture of accessibility requires an organization to actually be proactive.

Now, your organization may be thinking about how to implement accessibility into your current web design process. And different organizations may have different stakeholders contributing to the design and the development in different ways. Now, these resources may be all in-house, or they may be all outside vendors, or they could be a mix of both. And having worked with different customers, we've developed a strategy that cuts across all these differences and makes accessibility much simpler.

It all revolves around splitting up the work and providing each of these groups the resources that they need to easily understand what is required to create accessible content based on the specific role that they play in that process. So what's the secret to making web accessibility simpler and better? Well, it really comes down to two words, and that's "better requirements." Now, WCAG can be very vague. And this may seem like a simple thing to accomplish, but it really does involve a lot more than you might think.

First, it's a matter of how you define those requirements and for whom. And second, it's a matter of how you allocate and then deliver those requirements. Now, if you do these things right by following the strategies that we'll highlight today, web accessibility really does just simply fall into place. This approach that we're going to recommend, it actually has two parts. And those parts are defining the requirements and then delivering those requirements.

Now, both of these parts include two steps each. The first step involves breaking the vague and confusing WCAG Success Criteria into bite-sized elements that we call "core concepts." And the current gold standard for most that are achieving or looking to achieve accessible content is conformance with WCAG 2.1 level A & AA. Now, because that criteria can be vague, confusing, and often overlaps with other criteria, well, the first step is creating a clearer understanding for exactly what each of these success criterion requires.

Step 2 is breaking those concepts down further into interpretations, or what we call our role-specific guidance for technical and non-technical team members both. And step 3 is dividing the elements from step 2 into a series of checklists for each contributor's role in the web content development process. And finally, step 4 involves creating training and other resources at the core concept level that helps to tie everything together. Now, here it's important that technical and non-technical team members see the same training module, as that helps to reinforce the interdependencies and the mutual expectations that these team members have on each other.

Well, let's start with step 1. How do we go about dividing the WCAG Success Criteria into actionable concepts? Well, this step involves reading the Success Criteria that's available and all the associated techniques very carefully, and then asking yourself, what's the main ideas that are contained here?

Now, if you engage in this exercise, it doesn't take very long to see that the WCAG Success Criteria generally fall into one of three different categories. First, there are the easy success criteria. Now, for instance, Success Criterion 2.4.2 simply requires that web pages have a meaningful title that describe its purpose or its topic. Now, this prevents screen reader users and others from getting lost when they have several web pages open. And they're also easy, these criterion we're talking about here. They're easy to understand and easy to implement. And so our core concept in this case would just restate the criteria's requirement.

Then there are the deceivingly complex Success Criterion, like 2.3.1, which prohibits content that flashes faster than three times per second. Now, this is to prevent life-threatening photosensitive seizures. But when you read the requirement closely, shades of red flashing are far more problematic than other colors. Plus, the size of the flashing content also makes a difference.

But even here, there is a difference between whether the content is viewed on a mobile device or on a spacious desktop monitor. Therefore, after lots of research, the core concept is written as "flashing content shall not flash more than three times in one second and occupy a space of more than 21,824 square pixels."

And lastly, there are the overly general success criterion, like 1.3.1. It's only 16 words long, and it reads, "Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text." Now, this primarily helps a screen reader user make sense of the organization, the relationship of the content on a web page that a lot of sighted users just take for granted and almost do unknowingly when they visually scan the page's content. Yet, to explain what this Success Criteria encompasses, the W3C provides an assortment of 50 different techniques that cover a myriad of different issues that are all affected-- or rather, that all affect the structure in a web page.

Now, on the screen is an example of how we break up 1.3.1 into seven meaningful core concepts that are covered by the Success Criterion. Now, these cover topics such as header structure, ARIA landmarks, tables, and grouping of form elements. Now, one example of this is our core concept 1.3.1.1, Header Structure, which simply reads as, "Structure and headers follow a proper hierarchy of header tags." So this is one facet or nugget of information contained in WCAG Criterion 1.3.1. And this is what we mean by breaking down WCAG into understandable core concepts, not only understandable, but actionable core concepts as well.

Now, if we were creating a more technical, or I should say, a more traditional software requirements document, we'd probably just stop at step 1. But instead, each core concept needs to be further broken down into role-specific guidance for it really to be useful to everybody. Now, the term "role-specific guidance" is really a misnomer, as there really are three very general categories of roles that we need to address.

First, pre-production teams, such as copy authors, graphic designers, and UX engineers, really have requirements that need to be restated in a non-technical manner. Now, for instance, there's little point telling a non-technical wireframe designer that an HTML div element doesn't accept the alt attribute, and so they should use an ARIA label instead. That just wouldn't make sense to them-- well, most of them, I should say.

Now by contrast, that kind of technical guidance is exactly what that second group, the production teams, would need when they're going about developing the code for these web pages. And lastly, our QA teams require guidance that is functional in nature and on the test procedure being used. And this can involve telling teams when a code inspection is recommended or when and how assistive technology should be used in validating those web pages.

Now, we don't have time today to dive too deeply into how all this is done, but on the screen are the role-specific guidance for core concept 1.3.1.1, Header Structure, which we talked about just a few minutes ago. And so just based on that, you can see the differences that each team, the different type of guidance that they might be given to help them accomplish their part in that, creating that accessible content.

Now, if you'll recall what was said earlier, the first two steps of our process define the requirements. And the third step begins the process of distributing those requirements. Now, this is a simple and somewhat fun exercise in dividing up the role-specific guidance that is created in step 2. And we do this by breaking it into a series of checklists that align with the specific job roles in the organization. And how you divide up the requirements may change based on your organization's structure. But nevertheless, we found that following the basic structure works surprisingly well across almost every organization. And if it does require any kind of modification, it's usually just a few minor tweaks.

Now, first, the non-technical pre-production requirements get divided into three checklists-- one for copy authors, another for UX designers, and a third one for graphic artists and media creators. Now, next up, the production requirements go straight to a checklist that the web developers will use. And lastly, the QA and testing requirements go straight to a checklist for the QA and testing team members to make use of. Now, we do have a whitepaper on this topic, and you'll be able to download that after this presentation. And in that whitepaper, we've included a working example of what a full checklist for graphic or media artists should look like.

Finally, the last and possibly hardest step involves creating training resources and other information specific to each core concept that we defined in step 1. Now, you may think that keeping training granular for each role-specific requirement is a good idea. Yet, we found that this is not really the case. Instead, it's important to build training at the core concept level that we did initially in step 1 and not at the role-specific level that was created in step 2. And there are a few reasons for this.

First, if you defined your core concepts correctly, you'll likely end up with between 70 to 80 specific roles. And explaining that course concept and the associated role-specific guidance that goes with it and the key technical requirements takes about 10 minutes on average. And this is well within the attention span of most people, so it's good encapsulized in a small time frame so that it's easy for those to really go through and get that information.

Now second, by having technical and non-technical team members go through the same training on a specific core concept, this way they understand the interdependencies and expectations between the teams. Now, for instance, copy authors will understand how alt text that they're writing will get folded into the final web designs, and web developers will understand how copy authors are being told to present that alt text to them within the new designs.

And third, maintaining one page for each core concept is certainly a lot easier than maintaining three pages for each core concept. And after all, it's easier to update 70 to 80 pages on a regular basis compared to 200-plus pages. So we want to make sure that this is something that will continue to be refreshed and updated as changes and new techniques come along.

Well, that's pretty much it in a nutshell. There is no better way that we found to really include accessibility into the development content lifecycle. And doing it the way we've just outlined it allows teams to more effectively build and maintain web accessibility. Now, that was a lot of information. So let's kind of look at it another way to visually represent how this process works, to kind of drive the point home. So we'll begin with the nebulous WCAG level A & AA Success Criteria. These are imprecise and confusing as a set of guidelines just left on their own.

So first, we need to break these Success Criteria into actionable core concepts, and these are the specific ideas that each criterion is focused. And this is where most traditional requirement documents, the efforts that put into them, would actually end. But instead, we further break those requirements down into role-specific guidance for pre-production, production, and QA teams. Then we divide those requirements into checklists. And finally, we create training and resources that brings these stakeholders back together as a team.

Now, hopefully, you agree that this approach that we've just outlined has many advantages over the way that we have approached web accessibility in the past. It's far more flexible and works across any web development methodology. And if you're using an automated testing solution or have periodic accessibility reviews done on your site already, this process works seamlessly with those solutions.

It's fast, and it's immediate. There aren't days wasted sitting in a training class that the attendees will leave and likely forget 80% of that content. Instead, each training resource ties directly to a very specific requirement. And this just-in-time approach gives team members exactly what they need to know when they need it. And it allows them to immediately apply what they just learned as well, thereby helping retain that knowledge.

It works across organizations of any size. For large organizations, this approach is very practical because you can first prove success with a smaller pilot project before rolling it out across your entire web development team. It also promotes teamwork and accountability. Each team member knows not only what they have to do, but also why it's important and how it fits in with what the other team members are doing.

Now, we have just outlined a proven and effective way of infusing web accessibility into your web content development lifecycle, which can launch the accessibility of your online content like a rocket, just as our image on the slide indicates with our businessman sitting on a rocket quickly climbing into the air. Now, admittedly, taking the time to put this level of effort into creating internal standards can cost hundreds of thousands of dollars. In fact, at the start of this presentation, I mentioned that we have done similar work for some large organizations in the past. And they were large enough to absorb this kind of cost, and it did cost them hundreds of thousands of dollars.

Now, if this seems out of reach for your organization, don't worry. In fact, it's probably the most single most effective and affordable thing that you can do for your organization's efforts toward accessibility right now. And the reason it is very affordable is because we've already created that process for you. So we'd like to introduce you to WebAlign. And if you'd like to find out more about this revolutionary approach to web accessibility, just visit our website at Converge Accessibility and select Products and Services, and then select WebAlign.

But again, with approach that we've just outlined and along with the whitepaper that we're making available to you, you don't have to use WebAlign. You can do this on your own if you so choose. But if that proves to be overwhelming, then please consider looking into our product WebAlign and see how it can help you.

Now, in summary, there are three things that we discussed today. Now, first, the importance of an accessibility statement and how that statement can help reduce your legal risk exposure. Second, where and how to begin establishing an accessibility baseline of your current online presence. And lastly, we discussed a method to easily incorporate accessibility into an organization's web content development lifecycle.

Now, we hope this helps create a simpler and more direct path to achieving and maintaining accessible content for your organization, and thereby reducing the legal risk exposure of your online presence. Just like our businessman highlighted on the image on the slide, all those confusing and twisting roads to web accessibility have now been smoothed out, and he has a more direct path to get there and to be successful.

Now, some of the links that we mentioned throughout this presentation, we want to make sure you get those, too. And one of the first ones is our company website, convergeaccessibility.com, and then the W3C's Accessibility Statement Resource. And we will go ahead and read that again. That's www.w3.org/WAI/planningstatements.

We also wanted to make sure that you have the link to get that Better Requirements for Web Accessibility whitepaper, which really outlines everything we just talked about today, but in a little bit more detail. And it gives you some examples of what some of those resources would look like as well so that you can go ahead and go down that path yourself if you so choose. And that's at convergeaccessibility.com/whitepaper13. And, of course, if you would like to look into more about what our WebAlign product offers, you can do that at our website as well, as we mentioned before.

So we do want to thank Sofia and Samantha at 3Play Media for hosting this webinar today. And Ken and myself, we generally appreciate your generosity at 3Play Media for allowing us to share our thoughts and ideas through your platform.

SAMANTHA SAULD: Thanks so much, Jeff. We can start taking questions now. As mentioned, please use the Q&A window to submit any of your questions, and we'll answer them live or ask live. So the first question is, "I am in higher education. Should our LMS feature an accessibility statement? How about the individual courses?"

JEFFREY SINGLETON: Well, that's a good question. So the LMS itself, well, I know some of the universities I've been working with, as they are procuring new products, they are requiring an accessibility statement specific to the product. So that really is a good idea because, again, part of the accessibility statement is to allow for users to get the help they need at the time they need it and also to provide feedback.

Now, one of the other things you can do with your accessibility statement as well is, if you do have certain issues that-- I should say accessibility issues-- that have yet to be addressed on that product or on that system, you can actually highlight those within the accessibility statement, and then at the same time, provide the various workarounds if there are any workarounds available. And that way, that saves somebody who is struggling or having some kind of issue, especially with assistive technology, from thinking that they're doing something wrong, because then they can see that it is documented in that accessibility statement. So it does save a lot of trouble that way for everybody.

Now, as far as per course, that's probably not something that's going to be easily maintainable, especially when you consider the number of people that are creating those courses. But at a minimum, that LMS probably would benefit from a specific accessibility statement.

SAMANTHA SAULD: Great. Thanks, Jeff. The next question is, "Is WebAlign a content management system?"

JEFFREY SINGLETON: WebAlign is not a content management system, but it does run on a content management system. It's more of a compliance resource. And that's really kind of how we started out with WebAlign. And as we've been working with our various customers, we're finding that they have a lot of different ways of using that.

One way is not just for their own team members, but we've seen where some of the procurement officers-- you require certain products to be accessible, but how do you go about proving that? Well, some of these have given the vendors access to their WebAlign account and said, these are checklists we need you to go through here and make sure you've accomplished these tasks. And then they also have access to the resources that explain why these things are important, and how to implement them, and how to test for them as well.

And so that way, if they do get their product after this has been done and there's still an issue, the customer could go back to that current checklist that was provided and hold that vendor accountable and say, hey, you said you did this, but you didn't do it. Now go back and fix it. And so it has provided a unique way in making sure that there's some measurable way of holding some of these third-party vendors accountable in that way.

And another thing about WebAlign, too, we're adding new features to it. Next week, we're launching a feature where you can create these checklists online for your group, and it's for your group only. And there's also a discussion forum for your group only as well so that you can have all that information and have that discussion all in one place.

And another thing we recently implemented, it's still in beta, but it's what we call a checklist generator. And so one of the things we find is that when somebody goes to actually test for accessibility, the amount of information there can be extremely overwhelming. So what we've done is we have a little wizard that goes through and asks you, what kind of content currently exist on the pages that you're testing?

For example, if you don't have any multimedia content, well, you check that off. And then when that checklist is generated, it provides you a list that shows those items as already completed. So you don't have to spend your time trying to figure those out. So it's kind of unique, and we've actually used it for some of our own projects recently. So that's come in very handy. So it's more than just-- it's not really a content management system. It's actually more of a resource, a compliance resource.

SAMANTHA SAULD: Awesome. The next question is, "What are the key differences between WCAG 2.0 and 2.1?"

JEFFREY SINGLETON: So WCAG 2.0 and 2.1, some of the key differences are additional-- there's one additional guideline that's been added. And I can't remember off the top of my head the number of criteria that have been added because I don't often think about the difference. But there is that difference. And so some of this new criteria that's been added in 2.1 addresses more of the mobility aspect, or I shouldn't say mobility, but more of the mobile device platform. For example, it's looking at things like the reflow and then other things in a cognitive nature, like line spacing and things like that.

So if you're just now starting out with WCAG, you don't necessarily have to start with 2.1, unless your organization is mandated that you do so or that you have some other kind of legal requirement to do so. So what I would recommend, if you haven't even started yet, shoot for WCAG 2.0 and get that solid under your belt. And then it's a lot easier to move over to 2.1 because some of the stuff that's been added in 2.1 isn't as always easy to implement or test for. And then this year, I believe they're planning on releasing WCAG 2.2, which adds a few more Success Criteria as well. So those aren't finalized yet. But once they are, we'll certainly be incorporating those into our core concept guides as well.

SAMANTHA SAULD: Great. Thank you. The next question is, "What are your thoughts on free web accessibility scanners?"

JEFFREY SINGLETON: So free web accessibility scanners, I think they're great tools. Well, I should say, let me clarify that. There are some really good ones out there, and I use them myself when doing various reviews for web pages for our different customers as well. But I don't rely on them. And the reason being, as mentioned in the presentation, is oftentimes the results you get are sometimes confusing in the sense that it's telling you there's a failure on your page, but you can't really identify where that failure is or why it's a failure.

And sometimes that's because what it's reporting is really false. It's kind of misreading what's really been rendered on the page. And that's common with the scanners anyways because the browser gets the information and renders the information. So if that scanner comes in and scans it before that rendering is fully complete, sometimes that can lead to problems and just cost you a lot of time with tracking things down and trying to figure out what's really a failure and what's not.

But as far as using them, there are a lot of scanners out there. But I pretty much stick with some of the primary ones, mainstream ones like WAVE. And then there is one that Microsoft put out. I believe it's called Accessibility Insights. I actually use that one quite a lot because that helps me to quickly identify the headers on a page and the navigation regions and things like that.

And it even gives you-- if you're new to testing for accessibility, it even gives you some guidelines on how to go about doing that as well. Unfortunately, it's not full-featured. It doesn't cover everything, but it's actually a pretty good tool. So as far as using those tools, I say certainly give them all a try. Find the ones that work for you, but just don't take everything they tell you as 100% factual. Do your homework and understand what it is you're really looking at and what you're trying to accomplish.

SAMANTHA SAULD: Thanks. The next question is, "Are there any key trends you are seeing from web accessibility lawsuits?"

JEFFREY Yes. So we've actually seen an increase this last year in those lawsuits. And

SINGLETON: unfortunately, the majority that we're seeing are what we call the professional plaintiffs, or the surf-by lawsuits. And what I mean by that is, there's individuals and law firms out there that recognize that it's pretty easy to identify websites that are inaccessible or have some accessibility issues. And then often, as mentioned, they often use freely available scanning tools to do that.

And once they find a site, what they're doing now is they're finding somebody who actually has a disability or requires some kind of accommodation to go look at that site and then confirm that, yes, I can't do this, or I can't do that. And then they, I think, cost $400 to file the complaint, and most of these lawsuits do settle before they ever go to court. So for that $400 filing, they end up making quite a few thousand in most cases.

As far as some of the trends that I am seeing is-- we didn't talk about accessibility overlays or plug-ins. We're seeing a lot of sites that are using those. But unfortunately, you-- matter of fact, I think the blog post we put on our website this week talked about that. There are so many articles out there that discount the use of those type of overlays and the problems that they have. So we didn't cover that in our article. But what we did cover in our blog post was the fact that using these overlays can expose you more so than not using the overlay because for a few reasons.

First, it shows that you're aware that you need to do something about accessibility. And instead of doing what you should do to conform with the guidelines, you've put this third-party solution on there that's cheap. And in most cases, it's cheap, not always. But that actually increases the level of effort and sometimes makes things more inaccessible on your website. So that's a bad thing.

And the other things that we thought about with those overlays is, there also could be a potential security concern there because now you're adding another company's script into your web pages. And if that company were to be hacked or something like that, we're not sure how that could impact the people that are using that script on their web pages. And then there's also the privacy aspect that comes into play. Most of these plug-ins and overlays don't provide any kind of privacy policy. So we're not sure what kind of information they may be tracking from the users coming to our site and how they're using that information.

So there's a lot of exposure there along with the inaccessible nature of those overlays. So we do encourage you to try to avoid those if you can. And in saying that, one of the things that I see starting to happen is a lot more lawsuits are highlighting the fact that an inaccessible site is using these overlays.

And so I believe-- this is my own personal opinion, but I believe that this year we might see companies start to target sites that are using the overlays. Because in most cases, if you're using an overlay, you probably didn't make the underlying site accessible in the first place, or you wouldn't need the overlay. And considering how problematic they are, they could make some very easy targets when it comes to some of these lawsuits, these legal complaints that are being filed.

SAMANTHA SAULD: Thanks, Jeff. The next question is, "How often--" I'm sorry. I asked that question already. Oh, actually, no, this is a new question. "How often should you be conducting accessibility tests?"

JEFFREY SINGLETON: Well, that's a good question. That really depends on the type of process you have in place. Now, if you're regularly updating your website, it really should be part of your core unit test before that information's ever released. But if you don't update your site that often or if you're going through development sprints that maybe, say, take three to six months, you should be doing it at critical stages, just like you might do if you're writing compiled software. You want to make sure that at key release stages, you're validating that accessibility through that testing.

At best, I would say you should do it-- if your site isn't being updated that frequently and you're not making that many changes to the design and things like that, I would recommend you do it at least once a year as far as a full review. You should be doing that as you're developing the content and, as I mentioned, making sure that you're doing that unit testing and that sanity testing before you ever put it on your site.

But as far as looking at your site, that's a good thing to do once a year. And oftentimes, we find a lot of organizations choose to use third-party vendors for that. And the reason that is is because if you get into a rhythm within your teams, you're going to start missing the same things over and over. By engaging with a third party to do that occasionally, they're more likely to identify those holes and where things are slipping through and creating more accessible content on your site.

Now, that being said, when it comes to testing, you definitely want to start-- the accessibility, everything, really should start at the beginning. You don't want to become reactive to everything. It really should start, as we've discussed in the presentation, that proactive approach. And the other aspect of that-- and I think I just lost my thought. But the other aspect to that is the web browsers are updated. Different technologies are updated throughout the year as well. So it is good to do that regular checking on your website.

And that's also where scanning solutions really start to play a key role in your process. Because once you have content out there and live on your site, especially if you have multiple content creators creating content, the framework of the site might be pretty strong and accessible. But as that content gets added, especially if new people come into the organization and people leave, you could start seeing inaccessible content, unique content being introduced. And that's where scanning solutions really give you a great bang for your buck because it allows you to monitor those pages on a regular basis and those changes.

And it also gives you a way not only to identify those situations where something has become inaccessible, but it also gives you a way of providing some kind of a standard reporting over a period of time so that you can kind of develop a trend line and say, hey, are we getting better, staying the same, or getting worse? And that also comes into play, too, when you get a legal complaint, because then you can reference those reports and say, listen, this is what we've been doing over the last three years, and this is our progress, and this is how we're reporting things. So you get that reporting aspect to it as well. So I hope that answered your question. I think I got off on a tangent there.

SAMANTHA SAULD: Great. Thank you. The next question is, "What should we keep in mind when it comes to mobile accessibility?"

JEFFREY SINGLETON: So mobile accessibility, one of the biggest things I would say to keep in mind is, they say if you're going to have a mobile presence or a mobile layout that it's best to develop that mobile layout and then build outward instead of building the desktop layout and building down. And I think that makes a lot of sense, but that's more of a design consideration.

But as far as reviewing your content, all the WCAG criteria still applies in that way, but really comes down to testing. One of the things I've seen go wrong with a lot of the testing of mobile content is using mobile emulators or even browsers like Chrome. And I think Edge provides it as well in their Developer Tools. They give you the ability to emulate different types of devices or emulate different type of layout sizes. And that's all well and good, but that becomes a problem if that's the only way you're validating that content is through the desktop, through an emulator.

So my recommendation is to always use an actual mobile device to make sure that your content is working in that mobile layout, because we've seen situations where some issues might be identified in the emulator or things might work in the emulator, and then when you get to the mobile device, that issue either doesn't exist, or you get an entirely different issue that only shows up.

So my opinion is, and one of the things we do when we do web audits, is we always test on the mobile device in addition to testing on our desktop layouts as well. And sometimes we incorporate tablet testing in that as well, if the client deems that it's-- if they have enough responsive design that's specific to the tablet, then we'll include that as well. And as I mentioned earlier, that manual testing often does require multiple test passes, and that includes testing on those mobile devices as well.

SAMANTHA SAULD: Awesome. Thank you. So we have one more question. Our final question is, "What tips and tricks have you learned along the way for getting budget or buy-in for accessibility?"

JEFFREY SINGLETON: Yeah, so that one's really difficult. And that's where, when we talked about the accessibility statement, that's one of the key things you want to do when you start the process of developing that statement is you want to get all the different shareholders involved. And that includes your legal team, because if you get sued, they play a part in that as well. So they need to know what the guidelines and the requirements are that you're trying to target and what you're doing about it.

And it's good to get your management team in there, too, if you can get somebody from management to sit in, because having that open discussion and everybody kind of agreeing that this is what we need to do as an organization. And now it's not just the developers or the designers or one person in the organization saying, hey, look over here, this is something we need to do. It becomes more of an organizational awareness. And at that point, it's a lot easier to get support to do that.

And one of the other things I guess I would throw out as something to try to avoid is throwing too much money out unnecessarily when you're trying to start out with that effort. And what I mean by that is, I've seen organizations say, hey, we're going to do something about accessibility. And the first thing they do is they go out and they buy some type of subscription to one of these enterprise accessibility scanning tools, which aren't cheap.

But then they never really learn how to use that tool properly. And the information that comes out is so overwhelming, they're not sure what to do with it, as we discussed earlier. Or they might go out and they might purchase a copy of JAWS, which is going to cost you a pretty good amount of money as well. But then they give that to somebody to figure out how to use this, but they never really fully understand the ins and outs of using that screen reader. Because the screen reader itself is not a testing tool, it's actually a tool to help users get to the information on a web page.

And so a lot of those assistive technologies will make assumptions when situations arise where something isn't necessarily accessible. For example, if you have a form with labels, and those labels aren't programmatically associated with the input fields, JAWS will often make assumptions of to what that label should be. And usually, because the most pages are laid out the same when it comes to forms, it's usually correct. But there are times when it's incorrect. But the thing to know is that JAWS is, at that point, masking a true accessibility issue that's going to impact other assistive technologies down the road.

So again, just picking the correct route to really educate your team. And I think the best way to do that is engaging with a third-party vendor that's a quality third-party vendor that's actually going to hold your hand and give you the information and the understanding of what these issues mean and what should be done about them is a big thing. And then again, something like going through this process of creating these core concepts, like we've done with WebAlign, is also another great way to do that because it breaks it down in such a way that it's not overwhelming to anybody. And that's probably some of the things I would keep in mind.

SAMANTHA SAULD: Great. That's all the time we have left. Thanks, everyone, for joining. And thank you, Jeff, for a great presentation.