SAMANTHA SAULD: Thanks for joining this webinar entitled A Better Way to Make and Keep Websites Accessible. I'm Samantha Sauld from 3Play Media, and I'll be moderating today. I'm joined today by Ken Nakata, Principal of Converge Accessibility. Ken is a former senior trial attorney with the US Justice Department's Disability Rights Section.

He developed nationwide ADA policies for the internet, was awarded the Attorney General's Award for Excellence in IT, served as lead counsel for the inter-agency working group, making the federal government's IT accessible to people with disabilities, and represented the US many times in litigation in federal courts.

Ken is a frequent lecturer, and he is sought after by major international accessibility conferences, separately for his knowledge of law and technology. And with that, I'll hand it off to Ken who has a wonderful presentation prepared for you all.

KEN NAKATA: Well, thanks, Samantha. I appreciate it. Good [AUDIO OUT], everyone, and thank you for attending. As Samantha said, my name is Ken Nakata. Some of you may know me for the legal updates I give on web and digital technology. But today, I'm going to do something different and tell you about a better way to make websites accessible and keep them that way.

To be respectful of your time, this presentation should only take about half an hour. And that should [AUDIO OUT] of time at the end for Q&A. So thanks, Samantha and Sofia, and 3Play Media, for the opportunity to present here today. It should come as no surprise that web accessibility litigation is on the rise. In 2015, we only had about 57 federal lawsuits that were filed nationwide over inaccessible websites.

By 2018, that number had grown to 2,258. That's almost 50 times as many suits in 2018 as in 2015, only three years earlier. In 2020, this year, we expect between 4,000 and 5,000 lawsuits. Now, this number fails to reveal the true [AUDIO OUT] of this problem, however, because, well, first, it doesn't include the vast number of cases that are filed in state courts, particularly in California.

And it also fails to reveal-- it also doesn't include the cases that are resolved informally before a lawsuit's filed. And it also doesn't include the complaints that are filed with administrative agencies, such as the Department of Justice or the

Department of Education. While we may never know the true number of complaints, it's reasonable to estimate that the final number is probably somewhere over 10,000 complaints each year.

Now, this is potentially hundreds of millions of dollars that companies are paying to resolve web accessibility legal actions each year. But this huge volume also belies two other facts. First, that many complaints means that there's a ton of inaccessible websites out there. And that illustrates how people with disabilities are disadvantaged in our online world.

During COVID-19, websites that are vital for basic goods and services. And so it's more [AUDIO OUT] now than ever that websites be accessible to everyone. I rely completely on online services now, and I can't imagine what I would do if I couldn't use them. But second, [AUDIO OUT] litigation [AUDIO OUT] poorly understood and [? difficult ?] to accomplish.

Today, we're going to quickly talk about a way to [AUDIO OUT] these problems, and that's by making it easy to build and maintain accessible websites. So web accessibility, of course, is usually measured against the Web Content Accessibility Guide, or WCAG, as we call it, which is created in an open-source process by the W3C. WCAG is divided into three levels of conformance-- A, AA, and AAA.

And meeting the first two levels, A and AA, has become regarded as the de facto safe harbor, if you will, for accessibility compliance through settlement agreements and consent decrees with the US Department of Justice. WCAG has also evolved over time. WCAG 1.0 was a really loose set of guidelines that wasn't enforceable. Version 2.0 added a lot more specificity.

Currently, we're on version 2.1. Nevertheless, WCAG 2.0 levels A and AA has already become an international standard that's used by most governments worldwide for web accessibility. And WCAG 2.1 A and AA is slowly gaining traction as the new replacement. It's an ever-evolving process, however, as WCAG 2.2 is right around the corner.

Fortunately, each 2.x version of WCAG just builds on the last one. So if you've already met one version, meeting a new version is just adding in a handful of new requirements. That also means that, if you meet WCAG 2.1 A and AA, for instance, you automatically meet version 2.0. So during today's presentation, we're going to talk mostly about, how do you build websites to conform with WCAG 2.1 A and AA?

So WCAG 2.1 is divided into about-- A and AA, are divided up until about 50 different success criteria. And they all follow a common three-digit numbering schema. And I think one of the reasons that web accessibility is so challenging is because these success criteria are written as guidelines and not specific engineering specifications. Plus, the WCAG success criteria are often written in a really general way. So a single problem can often overlap into different success criteria.

All of this isn't meant as a criticism of WCAG or to W3C. WCAG was never intended to be a rigid set of requirements and was, instead, always intended as a set of universal design guidelines. Nevertheless, several industries have developed around web accessibility and for making websites conform with WCAG 2.1.

Unfortunately, these approaches have largely been expensive or unproductive in helping organizations internalize WCAG 2.1. First, we have companies that produce overlays, or widgets, to improve accessibility by adding a button on your website. When companies like this promise to make your website compliant with WCAG 2.1, I think that verges on an outright lie.

If you don't know anything about web accessibility, it's easy enough to think that they improve accessibility when, in fact, they often make it worse. This is because people with disabilities commonly use their own assistive technology, like screen readers and screen magnifiers. And these overlays often interfere with how that assistive technology works. To my mind, companies that make such claims are even more harmful than the plaintiffs' attorneys who sued the companies in the first instance because they're selling an ineffective solution to a desperate client.

Next, we have automated testing solutions. Now, these are produced by pretty [AUDIO OUT] companies. And their sales team will, or at least they should, be telling you that automated testing can only scan for about 25% to 30% of the WCAG requirements. Well, it just so happens that those 25% or 30% are some of the most important requirements. But it's not really enough to protect a company.

The problem, of course, is that so much of web accessibility involves features in the user interface that not even the smartest programmer can build a perfect rule to catch. Just to take some obvious examples, how can a computer program determine if a table is being used to really present tabular data? Or is that table just being used to achieve a pretty layout? Or if you've got a format, like a checkbox, how can a computer program really reliably determine which text near that checkbox needs to be programmatically associated with that checkbox?

These are absolutely vital accessibility requirements, but they're not easily reduced to code that a computer can test. And really, it's that unpredictable nature of web pages that makes overlays, that we talked about earlier, fail. The only real way to test a web page is through manual testing. But that's extremely labor intensive, requiring between 20 and 45 minutes per page. Obviously, that means that companies can only afford to manually test a handful of pages.

So all of these solutions are backward-looking, however. A forward-looking approach offered by these same accessibility companies is training. But while that's useful, training usually only focuses on development. And even when the training reaches beyond developers, like to designers and to QA teams, training tends to be a random laundry list of best practices.

One thing that never seems to be fully done is [AUDIO OUT] single vision of accessibility between designers, developers, and QA testers. So before we get to the solution, we need to talk about who can benefit from these ideas and also who can't. I don't want to be wasting your time during this presentation.

Of course, I suspect you wouldn't be here if you either didn't have some role in or weren't interested in web development. But that includes a huge swath of people, and not everyone's going to benefit from these strategies. So as I talk about who can benefit from this part of the discussion and who can't, ask yourself if you find yourself in one or more of these categories.

First, and most obviously, we have any organization that has an internal web development team. This can be a public-sector agency, a not-for-profit, or a for-profit company that has a reasonably sized team of people that focus on UX design or code development and QA testing.

Next, we have organizations that outsource their web design and development to outside vendors and design agencies. The strategies I will talk about today help ensure that your outside vendors take accessibility seriously and that accessibility is being incorporated at each stage of the process.

Last, we have any organization that struggles with ensuring that web accessibility is really understood by everyone who's involved in that web development. An organization can have extensively [AUDIO OUT] requirements. And that doesn't necessarily mean that everyone on the team has a single vision of accessibility and really understands what their role is in fulfilling that vision. Oddly, this can be true for even really big organizations that have their own accessibility standards to meet global markets.

Now, as I said, there are some organizations that won't benefit from this presentation. And that includes any very small company or organization that develops its own websites and content management systems, like Wordpress or Shopify. So if you're a [AUDIO OUT] proprietor creating a Wordpress site on your own using Beaver Builder and installing various plugins to provide functionality, this presentation will likely have very little use for you unless you're really competent in development skills and don't mind creating your own child themes and custom plugins.

On the other hand, if you're hiring an outside agency to design and build your WordPress or Shopify site, then absolutely these strategies will be relevant to you. But they'll be even more important to your WordPress or Shopify developer. So let me provide a little bit of quick background because it segues into explaining how I came to realize that this is a better way for creating accessible websites.

So as Samantha said, again, my name is Ken Nakata, and I'm the principal and co-founder of Converge Accessibility. We're a new accessibility consulting firm since about June of this year, so we're really young, started by me and my colleague Jeff Singleton. Even though we're a new company, we've both been heavily [AUDIO OUT] in the accessibility community for the last 15 years.

We've presented at conferences like CSUN and M-Enabling and Accessing Higher Ground, the Pac Rim Conference on Diversity and Disability more times than we can remember. And we've helped companies, large and small, with their accessibility projects. I think I'm most proud of being the only consultant in our space to [AUDIO OUT] NASA to help make sure that their grantees, such as museums and science centers, included accessible technology for tomorrow's [? generation ?] of scientists and engineers.

But my story starts long before I became a consultant. Before joining the private sector, [AUDIO OUT] I was a senior trial attorney with the US Department of Justice. I spent 12 years working in the department's Disability Rights section, and I had a fantastic career, starting with built environment accessibility litigation and moving into digital technology accessibility.

While I was there, I was the lead counsel for the interagency Section 508 working group, helped develop the Access Board's Section 508 rule in technical assistance, and created the department's technical assistance under the ADA for websites. In fact, some of the technical assistance I wrote while I was there, such as the accessibility of state and local government websites for people with disabilities is still posted on the government's ada.gov website.

The solution I'll talk about today has quite a history, starting back with my work at the Department of Justice. When I was back at DOJ, I transitioned from litigating traditional ADA cases to focusing on and developing the Section 508 regulation and standards around 1998 and 1999. Now, as you may know, these are the rules governing the accessibility of what the federal government can buy.

I helped write the original Section 508 standards, and I also wrote the technical systems for how to code accessible websites, which is still available on the Access Board's website. After leaving DOJ, I started working as a consultant. A few years later, in 2001, I was approached by one of the largest software companies and asked to help write their global accessibility standards.

And I suppose word of my work spread because, four years later, one of the world's largest IT companies asked me to do exactly the same thing, except, this time, for hardware and software. Now, both of these companies are enormous. They each have over $100 billion in annual revenue, and they have over 100,000 employees worldwide. They each needed a [? terse ?] set of internal standards that they could use with the hundreds of product teams that they worked with around the world.

Then, in 2016, I was asked by a much smaller IT company-- they were puny. They only had $4 billion in annual revenue and over 9,000 employees worldwide-- to do exactly the same thing. Except this time, they wanted me to include non-technical creative staff and to design a set of requirements that was flexible enough for a rapid, agile development process.

Now, each experience taught me valuable lessons along the way. From working with the Access Board and creating the Section 508 regulations, I discovered the need for precise, legally enforceable standards. At the time, WCAG 1.0 was the norm for websites. I knew right away [AUDIO OUT].

From working with the large multinational corporations, we learned the importance of creating requirements that focused on developers and QA personnel. One set of requirements needed focus separately on developers and QA teams. But I think the most valuable lessons were brought home to me from working with the last client, the small $4 billion IT company.

Here, we learned the importance of parsing requirements into different levels that included technical and non-technical [AUDIO OUT] members. But while all these companies may seem large, the strategies that we learned applied just as much to a large organization as a small one. So let's talk about what that strategy is and how you can do it.

So what's the secret to making web accessibility simpler and better? It really comes down to two words, better requirements. Now, before you just jump off the line in frustration because I'm repeating something that you already know, I'll ask you to bear with me because those two words actually involve a lot more than they may first appear.

First, it's a matter how you define those requirements and for whom. And then, second, it's a matter of how you allocate those requirements and then deliver those requirements. If you do those things right, by following the strategies we'll talk about today, web accessibility really does just fall into place.

So the approach that I recommended involves two paths, defining the requirements and delivering them. Both of these parts involve two steps. Step one involves breaking down the vague and confusing WCAG success criteria into bite-size elements that we originally called universal requirements, but that we now call core concepts. As I mentioned earlier, the gold standard is conformance with WCAG 2.1 A and AA.

Because those success criterias in WCAG 2.1 A and AA are vague, confusing, and overlap, the first step is creating a clear understanding of exactly what each success criteria requires. Step two is breaking those elements down further into interpretations or what [? I call ?] role-specific guidance for technical and non-technical team members.

Then, step three is dividing the elements from step two into lists for each job role in the web development process. And then, finally, step four is creating training and other resources back at that core concept level in step one that ties everything together. Here, it's important that technical and non-technical team members [? receive ?] the same training module, as it reinforces the tendencies and mutual expectations that team members have on each other.

So let's start with step one. How do you divide up the WCAG [AUDIO OUT] criteria into actionable concepts? And what do I mean by that? Well, the first step involves carefully reading the success criteria and all of the associated techniques very carefully and asking yourself, what's the main idea that's contained here?

Now, if you engage in this exercise for very long, it won't take very long to see that the WCAG success criteria [AUDIO OUT] one of three categories. First, you've got the easy ones, the easy success criteria, like success criteria 2.4.2, which requires that web pages have a meaningful title that describes its topic or purpose to promote screen-reader users and other people with disabilities from getting lost when they have several web pages open. OK, these are easy, and the core concept can just restate that.

Then, there are the deceiving complex success criteria, like 2.3.1, which prohibits content that flashes faster than three times per second. 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 makes a difference.

But even here, there's a difference between whether the content is being viewed on a mobile device, where it may be held really close to the eye, and a spacious desktop monitor where the screen will be pretty far away from the user's eye. So that's a [INAUDIBLE] digging through the interpretations and the guidance.

2.3.1 developed a core concept to [AUDIO OUT] be written as flashing content, flash more than three times in one second, and [AUDIO OUT] space of more than 21,824 square pixels. Lastly, there are the overly general success criteria, like 1.3.1. Here, that success criteria is only 16 words long. And it reads, "information, structure, and relationships conveyed through presentation can be programmatically [AUDIO OUT] or are available in text."

Now, this primarily [AUDIO OUT] the users makes sense of the organization of a web page, [AUDIO OUT] users take for granted [AUDIO OUT] unconsciously when we visually scan a page's content. Yet, explain what that success criteria encompasses. The W3C provides an assortment of 50 different techniques that cover a myriad of different issues that all affect the structure of a web page.

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 criteria. Now, these cover topics such as header structure, ARIA landmarks, tables, and grouping of form elements. One example of this is 1.3.1.1, Header Structure, which reads simply as, "structure in headers follows a proper hierarchy of header tags." So this is one facet or nugget of information contained in 1.3.1. And this is what I mean by breaking WCAG down into core concepts.

Now, if we were creating a traditional software requirements document, we'd stop right there at step one. But instead, each core concept needs to be broken down further into role-specific guidance. The term role-specific guidance is really a misnomer because there are really only three general categories of roles that we need to address.

First, we've got preproduction teams. And that includes people like copy authors, graphic designers, and UX engineers that need requirements restated in a very non-technical way. For instance, little point telling a [AUDIO OUT] that div elements don't accept the alt attribute, and so they should instead use ARIA-label instead. By contrast, that kind of [AUDIO OUT] guidance is exactly what's needed with the second group, production teams, when they're writing code for their web pages.

And lastly, QA teams need guidance that's [AUDIO OUT] in nature based on the test procedure that's used. And this can involve telling teams when a code inspection's recommended or when and how assistive technology should be used. So we don't have time today to dive too deeply into how this is done. But on the screen, are the role-specific guidance for 1.3.1.1, Header Structure, which we talked about a few minutes ago.

If you would like a copy of this, it will be in the white paper which you can download later. Now, if you recall what I said earlier, the first two steps in this process define the requirements. The third step begins the process of distributing them. Now, this is a simple and somewhat fun exercise of dividing up the role-specific guidance created in step two into a series of checklists that align with the specific job roles in an organization.

Now, how you divide up the requirements may change based on the specific nature of your organization. But nevertheless, we found that the following structure works surprisingly well across almost every organization with just a few minor tweaks. First, non-technical production-- preproduction requirements get divided into three checklists-- one for copy authors, another one for UX designers, and a third one for graphic artists and media creators.

Next, the production requirements go straight to a checklist for web developers and programmers. And lastly, the QA and testing requirements go straight into a checklist for QA and testing team members. And in our white paper, which, again, you can download after this presentation, we've included a working example of what a full checklist for a graphics or media artist should look like.

Finally, the last and possibly the hardest step involves creating training resources and other information specific to each core concept that we defined in step one. While you might be tempted to think that it's a better idea to keep the training as granular as possible for each role-specific requirement, we found that that's not the case.

Instead, it's important to build training into the core concept level, which we created in step one, and not the role-specific level created in step two. And there are a couple reasons for this. First, if you define your core concepts correctly, you'll likely end up with between 70 and 80 specific rules. Explaining a core concept, the associated role-specific guidance, and the key technical requirements [AUDIO OUT] 10 minutes on average. And that's well within the attention span of most people.

Second, by having technical and non-technical team members go through the same [AUDIO OUT] on a specific core concept, they understand the interdependencies and expectations between the teams. For instance, copy authors will understand how the alt text that they write will get folded into the final web designs. And similarly, web developers will understand how copy authors are being told to present that alt text to them in the new designs.

Third, maintaining one page for each core concept is certainly easier than maintaining three pages for each one. After all, would you rather maintain 70 to 80 pages or 210 to 240 pages? So that, in a nutshell, is a better way to build and maintain web accessibility.

Here's another way to visually represent this process in a much shorter, simpler way. We begin with the nebulous WCAG A and AA success criteria. These are, again, an imprecise and confusing set of guidelines. So we first need to break the success criteria into actionable core concepts.

These are, again, the specific ideas that each core concept is driving at. And this is where most traditional requirements documentation efforts would end. Instead, we need to further break those requirements down into role-specific guidance for preproduction, production, then QA. Then we divide those requirements into checklists. And then, finally, we create a training and resource page that brings the stakeholders back together [AUDIO OUT].

So now, I think you'll agree that it should be obvious that this approach has a lot of advantages over the way in which we've approached web accessibility in the past. First, it's far more flexible and works across any web methodology. If you're using an automated testing solution or you have periodic stability reviews done on your website, then this process works seamlessly with those solutions.

It's also fast and immediate. There aren't days wasted in training that's quickly forgotten. Instead, each training resource ties directly to a very specific requirement. And so this is what I'm going to call, just-in-time approach, gives team members exactly what they need to know when they need it. It also works across organizations of any size.

Or for large organizations, this approach is very practical because you could first prove success with a smaller pilot project before rolling it out to your entire web development team. And 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 other team members are doing.

But I know what you're thinking. Ken, those projects where you created these internal standards cost hundreds of thousands of dollars? And well, we can't afford that. Well, in reality, it's also extremely affordable. In fact, it's probably the single most affordable and effective thing you can do for web accessibility. OK, so I'm not a natural salesperson, so I'll keep this part brief.

The reason it's actually very affordable is because we've already created it for you. So we'd like to introduce you to WebAlign, which we offer on a per-seat subscription basis for web development teams of five or more team members. Now, even small teams that have never even heard of web accessibility can immediately get a handle on it and have the same kind of web accessibility program as some of the best IT companies in the world.

To create WebAlign, we completely rewrote and expanded the core concepts to derive from each core success criteria and then refined the role-specific guidance to be useful across all organizations. We also developed five ready-to-use checklists for copy authors, graphic and media creators, UX designers, developers, and QA team members. And those five checklists address the needs of most organizations, but, of course, they can easily be changed if necessary.

Plus, each core concept is supported by professional-quality, fully captioned short training video. We've got over 70 of them in total and a page of resources that team members can refer to as they're completing their checklist. Each requirement on each checklist includes a link directly back to the page at WebAlign that discusses exactly what needs to be done so your team has the exact tools that they need at their fingertips. Plus, each video is only about 10 minutes long. So they can easily understand the requirement rapidly, meet it, and move on.

So there you have it. We firmly believe that the best solution is to have your own accessibility expert [AUDIO OUT] create your own requirement's library to meet your needs. [INAUDIBLE] can cost over $300,000 to create and hundreds of thousands of annually to keep up. If that's a bit beyond your budget, we created WebAlign, which is very cost effective and scalable. The smallest package starts at only $600 a seat, and the prices drop from that point as your team size grows.

It's completely a turnkey solution. The videos and checklists are already created. And it gets everyone immediately on board with a common vision of accessibility. Perhaps, best of all, we'll meet with your team weekly to make sure that everyone understands exactly what to do. We found out that a little bit of coaching and support makes a big difference.

Plus, your feedback helps us improve WebAlign for everyone. We've been working in web accessibility for over two decades. And WebAlign represents the culmination of years of experience and the best practices we've learned all along the way.

It took us almost 2,000 hours of dedicated effort to create. We really believe that, if you're serious about web accessibility and you can't afford to build your own system like the one I described today, you owe it to yourself to check it out. So now there's no excuse for not having your own state-of-the-art web accessibility program.

If you're interested in getting started, contact us at info@convergeaccessibility.com to subscribe to WebAlign. As a special thanks for attending today's webinar, we'll be sending an email to everyone that includes a link to a free downloadable web paper that presents actual examples and a full checklist for web requirements. You can use this white paper, either for creating your own web accessibility program or for getting the buy-in necessary to give WebAlign a try.

You can also visit our website and sign up for our popular newsletter. And while you're there, please like us on Facebook and follow us on LinkedIn. While it may seem like an irrelevant gesture, it actually means a lot, which is a strange thing for someone like me, who utterly hates social media, to be saying.

And if you'd like to discuss this presentation or anything else, please don't hesitate to reach out to me at ken.Nakata@convergeaccessibility.com. And again, thank you Sofia, Samantha, and 3Play Media for hosting this webinar. Jeff and I both have partnered with 3Play Media for years. And you guys really are leaders in the captioning space.

And now that we're a new company, you're helping getting our message in front of new friends and new clients is invaluable. So we really appreciate that. Thank you.

SAMANTHA SAULD: Awesome. Thanks so much, Ken. We can start the Q&A right now. We have a few questions. But feel free to type in any more questions that you all have in the Q&A box. So the first question is, what are the most common website accessibility errors that you see?

KEN NAKATA: Wow, gosh, it's really the basic things. It's things that are that are even [INAUDIBLE] with an automated testing tool, like missing alt attributes on images, missing [? formal ?] labels, those things are pretty easy to catch and are essential. But they're the first thing also that a plaintiff's attorney looks at. And yeah, they're just everywhere.

SAMANTHA SAULD: Great. The next question is, which industries do you think need to improve their online accessibility? And how do you think that change will happen?

KEN NAKATA: Well, personally, just a gut feeling, I think that e-commerce and health care are the two big targets right now-- e-commerce, because of COVID, health care because there's a lot of initiatives within the government to make sure that all health care systems are accessible. Plus, those are really the most essential needs that I think people have in our society. The move towards health care accessibility, I think, is being pushed, again, mostly by the government. But it's also having trickle-down effects.

So for instance, we were contacted by a hospital system in the Midwest that was evaluating different client portals. And I think that, indirectly, that comes from HHS. But ultimately, it's being pushed also, at a lower level, by hospital systems and health care providers. And then, e-commerce, I think that that's largely been driven by the plaintiffs' attorneys.

SAMANTHA SAULD: Great, thank you. The next question is-- great presentation, can you speak a bit more about pricing? You mentioned $600 a seat. Is that annually or monthly?

KEN NAKATA: Oh, that's on a six-month basis. And it's per seat, so it's really intended for everybody in the whole web development environment. So your UX designers, QA testers, and developers should each get a seat because they each have a different role.

SAMANTHA SAULD: Got it, thanks. The next question is, are there specific fonts that are well-suited for accessibility?

KEN NAKATA: Ah, fonts. No, there are not, but there is generally a tendency towards promoting sans serif fonts. I remember, after the last CSUN, I attended a really interesting session with a guy who was a font geek. And his entire life was spent around designing new fonts. And he was presenting some really interesting research on fonts that were more ideal than others.

So he contacted me after the presentation-- I may be able to dig up the information that I had on that presentation for best practices in font design.

SAMANTHA SAULD: The next question is, what do you think about, and do you support or offer, user testing by persons with disabilities or assistive technology users?

KEN NAKATA: That's a really great question. So there's two things here at play. There's usability, and then there's accessibility. And I like to draw a distinction between those two. Accessibility to me is meeting a set of standards, universal design standards that facilitated accessibility. That doesn't in any way diminish the importance of usability testing for people with disabilities.

And a lot of people argue that meeting those standards, like WCAG, doesn't fully address the needs of people with disabilities. And I completely agree with that. The problem is that it's hard to reduce usability to a set of rules. So I think that what has to be done is an organization first has to worry about compliance. And then it has to go the further step of improving usability.

And so once you get past all this stuff that I've been talking about, then I think it really becomes important to think about usability studies for people with disabilities to make sure that they're not going through a checkout process, for instance, that's twice as long as everybody else. That's not fair.

But unfortunately, again, you can't really reduce it to a set of enforceable rules. So yeah, usability testing, I think, is vital. But right now, I think what I'm talking about is just preventing a lawsuit.

SAMANTHA SAULD: Thanks, Ken. The next question is, what should you do when you have team members who are hesitant to change their ways and adopt accessibility standards?

KEN NAKATA: I think you have to make it easy for them. I think that it really comes down to-- you can't take, say, for instance, WCAG and just throw it in front of a developer. They will kick and scream and bite to prevent trying to adopt it.

Because, again, it's a vague and confusing set of requirements. The first time I looked at it, it took me at least a day to just to figure out how it's structured before I even thought about the different strategies for actually creating accessible content and how you code accessible content.

So that's part of it. So that's, I think, one of the reasons why-- that was one of the driving forces for us when we were creating this whole process of WebAlign was, we wanted to create rules that were understandable for different roles. Because a wireframe designer or a graphics artist has entirely different needs than a person who's writing code. And so if you can make it easy for them, and you can give them all the resources, then the resistance goes away.

SAMANTHA SAULD: Great, thanks. The next question is, going back to typography, wouldn't you think serif fonts are better for accessibility, as they have more elements to make the letter recognizable? And, for the most part, we read shapes, not letters.

KEN NAKATA: Yeah, I agree. But I remember-- well, first, most of the requirements in WCAG reference sans serif fonts. And I can't tell you exactly why. And again, I would ask that we contact-- contact me offline about this. Because this person that I was talking to back at CSUN-- I just remember having the most fascinating discussion about those kinds of issues.

Yeah, the things on serif fonts, for instance, where you've got the extra little elements that make up the header, the top of the font, and the bottom of the font, I agree. But they can also be visually distracting. And yeah, it was just a really interesting discussion. I'm no expert in that area by any stretch of the imagination, but I know somebody who is.

SAMANTHA SAULD: Thanks, Ken The next question is, does your platform help users keep track of their progress as they build or remediate a site? Is there any reporting the platform can deliver for stakeholders or project managers?

KEN NAKATA: No, we didn't build it as a specific platform. It's really just a set of checklists, at the end of the day, that tie in to the specific training and guidance. And it's really more a matter of delivering the guidance. So the checklists are really useful. The checklists actually weren't my idea. They were that last client that we had-- this [INAUDIBLE] IT company.

It was actually their idea to create the checklists because they wanted to put the checklists into their own system for tracking compliance. And when you've got these large companies, like the really big ones-- I don't want to name names, but it may be obvious who they are-- they have their own systems. And they wouldn't accept us trying to create our own system anyways.

SAMANTHA SAULD: Thanks. So we have time for a few more questions. So keep them coming in Q&A tab. The next question is, do you have resources for making PDFs accessible and marketing content?

KEN NAKATA: No, we don't do PDFs. We partner with companies that do PDFs. But it's really a matter of expertise. For us to make a PDF accessible would take twice as long as the other guys. And frankly, they do a better job because, well, that's all that they do.

SAMANTHA SAULD: The next question is, which standards should we be following-- WCAG 2.0 or WCAG 2.1?

KEN NAKATA: I think 2.1, because it's backward-compatible. 2.1 includes a couple of things that are a little bit-- well, not web-related in some senses. It's odd because there are things, like 2.1 has requirements for orientation, for instance. So for instance, if you're viewing content on a smartphone in landscape mode, and it only supports landscape mode, and it won't support portrait mode, that's a problem under 2.1. And that wasn't a problem under 2.0.

But when you look at the WCAG techniques that are associated with that, it's really hard to discern any kind of techniques. I suppose, with style sheets, you could do it by changing your viewport size. But in general, it's not something that's discussed very much in the techniques. I think that most of the things that are in there, though, most companies do anyways.

And if you meet 2.1, you automatically meet 2.0 because of the numbering schema. Basically what 2.1 did was, if the success criterion in 2.0 ends up with-- I'm just getting numbers off the top of my head-- but say WCAG 2.0 stops at 1.3.6, then WCAG 2.1 would pick up on 1.3.7 and go forward.

So it still includes all the things that came before it. So the numbering schema really makes a lot of sense. If you follow 2.1, you'll know exactly what you did for 2.0, and you'll future-proof yourself. So I think you might as well start-- shoot for the highest one possible.

The thing that gets a little bit confusing in that, though, is 2.2 is right around the corner. And the draft is already out. But I wouldn't recommend going for 2.2 yet because it's still in draft mode. The language could very well change, and the techniques obviously aren't there.

SAMANTHA SAULD: Got it, thanks. The next question is, if you are a large organization that has never made its website accessible, where do you start?

KEN NAKATA: Well, I know a certain product you could buy that would really, really help. Seriously, the last company is a good example of that. When I say they're puny, they're not puny. I mean, $4 billion and 9,000 employees around the world is a pretty decent-sized company. And so they had their own web development team internally.

And the system that we created was really based upon what their project managers needed because they didn't want something that was heavyweight. They wanted something that they could very quickly implement across their team and that didn't look like most requirements.

During the presentation, I talked about drilling down to requirements in step one and two, something that-- what I call core concepts. And I said, well, that's where most requirements efforts end. Actually, that's not quite true. Most requirements efforts don't even get that far.

So if you can create a very specific set of actionable requirements, like the core concepts, and then build out from the model that I built out on, I really think that if you're-- I really think that this is a model that is tailor-made for large organizations-- less so for medium-sized organizations. And the smaller you get, the less it applies. But it still has a lot of lessons for smaller organizations too.

SAMANTHA SAULD: Awesome. Thanks so much, Ken. Those are all the questions that we have for today. So I'd like to say, thank you to everyone for joining us. And thank you again, Ken, for an awesome presentation.