Transcript: Understanding Compliance Baselines and Implementing Accessibility Initiative, Ken Nakata. AccessibilityPlus 2022.

Good afternoon, everyone, and thanks for taking the time to join my presentation. My name is Ken Nakata. Normally I talk about legal issues to come up in web accessibility litigation, but with three lawyers in the line of drawing straws about what to talk about, I fell back on my consulting background to discuss how to make actually make web accessibility happen. I wouldn't be a typical lawyer if I didn't say a few words about myself. So before joining the private sector, I was a senior trial attorney with the U.S. Department of Justice. I spent twelve years and the department's disability rights section and had a fantastic career, starting with built environment accessibility litigation and moving into digital technology accessibility. I got to lead cases on some of the most important ADR cases and mental health negotiated settlement agreement that required Greyhound bus lines to provide accessible busses across the industry a year before today required them to.

And that provided accessible bus service to over 3500 travelers with disabilities and even negotiated the Justice Department's first consent decree for patterns or practices of police misconduct on a short detail to the special litigation section. While I was at justice, I was also the lead counsel for the entire Section 58 working group to helped develop the Access Board, Section five related regulations and technical assistance, and created the department's technical assistance under the ADA for websites. In fact, some of the technical assistance I wrote, such as the accessibility of state and local government websites to people with disabilities, is still posted on the government's Ada Dot gov website.

Now, my business partner wouldn't be happy with me if I didn't mention we've been doing what we've been doing since I joined the private sector for the last 15 years. I've worked with organizations to make their technologies accessible.

We've worked with companies like Microsoft and HP to refine their global accessibility standards and organizations like Nassr to help ensure that science centers and museums were accessible to tomorrow's generation of scientists, doctors and engineers, but most relevant to today's presentation.

We also worked hard to make web accessibility easier for organizations to understand and implement. And it's that portion of our work that I'd like to talk about today and how organizations can manage web accessibility. Now I'm going to jump right in with a terribly uncomfortable truth, even though a lot of plaintiff and defense counsel require a quick A and Double-A compliance as part of their settlement agreements. It's almost impossible to fully comply with those standards. first, we're CAG is impossibly vague and experts often disagree. What it means takes success criteria one point 3.1, for instance.

This is a level requirement and says information structure and relationships conveyed through presentation can be programmatically determined or are available in text. In a nutshell, this means that structures like tables and visual headings that are visually presented on the screen are also revealed to a screen reader.

We can make one point 3.1 is absolutely vital to accessibility and without it. Screen reader reader users would not be able to read information and columns or tables unless they possessed an extraordinary photographic memory. So a high level compliance is one point 3.1 makes perfect sense.

Unfortunately, at a granular level, it's almost impossible to always tell if a web page complies with one point 3.1 because work never tells you exactly what you have to do. Instead, the supporting techniques gives readers 50 different ways to meet it in 13 different ways to fail.

This means that while it's easy to identify clear examples of web pages that fail KEGG one point 3.1 It's much harder to say whether a specific supporting feature is necessary for compliance. For instance, two techniques Aria eleven and Aria 13 state that identified regions on a web page support would take one point 3.1, but they don't definitively say that they're required. While the ARIA specification and a new HTML five specification make it super easy to identify the structure of the different regions in a web page, accessibility experts argue all the time about whether we're OK at one point 3.1 really requires them.

For the record, I believe the work at one point 3.1 does require them, but I base that on the belief that on the fact that it's so easy to do and not because we take one point 3.1 says so.

Second, we includes requirements that can be ignored on web pages, and yet they remain completely. Accessible. We can make one point 3.5. So a requirement and it requires that any input field that calls for common information, for example, the user's last name also includes an auto auto complete attribute that identifies as fact in a common computer readable way. So our last name field would need to include an attribute that reads auto complete equals open quote family hyphen name close quote. Unlike week one point 3.1. This is very straightforward, simple and clear as a requirement. If your input field asks for a specific standard, information like a user's last name commonly falls on the falls on a list maintained by the W3C. Just make sure to set your autocomplete attribute to equal the value on the list. The problem People with disabilities aren't affected in the slightest. If you meet this requirement or not, so why is that?

The idea, which I one point 3.5, is brilliant. By standardizing on a list of common values, all users would be able to fill in forms much more easily. Plus, it helps users with cognitive disabilities because it wouldn't matter whether the last name field was visually labeled on the screen as last name or surname or family name.

It all just gets filled incorrectly. The problem is that this requires support from the browsers. For example, Edge, Chrome and Safari, and none of them built in support for autocomplete. Instead, most browsers use their own text recognition technology to fill in forms for users, and you've probably noticed that it makes a lot of mistakes which wouldn't have happened if they supported the autocomplete attribute properly. The bottom line is that you can very easily fail or completely ignore at one point 3.5, and it won't make the slightest difference to any user with a disability. Nonetheless, it's easy to spot violations of like one point 3.5, and almost any automated scanner could find violations of at one point 3.5. So these two problems the vagueness of WCAG and its requirements for things that don't matter are really technical issues. The third reason why weak compliance is virtually impossible is more practical. Simply put, most websites are large and constantly changing.

It's hard enough to get a physical building in a physical building to get everything right in terms of accessibility. And that's what architects know exactly what they're doing. Unlike physical buildings in the real world, web pages in a digital world are constantly changing and are constantly being modified or changed by people who have little to no understandingof work eg. So when that copy author adds a new image without a good art attribute or that summer intern posts a video recording without captioning liquor compliance just went out the window. Despite these shortcomings, lawyers let their clients promise.

Or at least they used to let their clients promise for quicker compliance to AA and AA granted within the last year or two. I've noticed lawyers getting smarter about this. For plaintiffs for quicker compliance doesn't make sense. For instance, why should you try to require companies to make their websites meet with code one point 3.5?

If it has no effect on any users with disabilities for defendants, it's impossible with any website that constantly changes, or even moderately frequently that changes even moderately frequently to be absolutely sure that someone didn't add content that wasn't fully in compliance was working a way.

So some lawyers try to get around this by settling cases at a lower standard like WCAG level. I think this is a mistake. Yes, we can. Levels are intended to accommodate different levels of accessibility, but I think we're going to see a really lousy job at this.

Instead, I think that the levels reflect the advocacy skills of different members of the work, eg working group, for instance. I don't think many of us would ever want to ignore the requirement for basic color contrast on our web pages without good color contrast.

Users with impaired vision, including people with color blindness or people with relatively poor eyesight, wouldn't be able to read words on a web page. But even the most basic requirements for color contrast don't come in until worker level Double-A and success criteria one point 4.3.

Instead of promising full whack a and Double-A compliance, it's important to be flexible here and accept that even at the best of organizations, that the best any organization can do is to make best efforts towards complying with work and establishing the goal of constantly striving.

Earlier work, a compliance is important, and I think it's the first thing that should go into an organization's accessibility statement. Users with disabilities need to know that you take accessibility seriously. While any moderately sized website cannot ever guarantee that it will fully comply with work, and users should know that you're at least trying and accessibility statements important, I think it's the most important thing that any organization can do to improve accessibility, and it's also the easiest to implement. I've already described the first element of a good accessibility statement, and that is identifying and articulating the level of work that you're striving for and your efforts towards meeting.

It was always meeting that target. But the reason I think in accessibility statement is critical is based on the effective communication requirement and titles to and titles and three of the ADA, which requires organizations to provide auxiliary aids and services to ensure effective communication.

Unless doing so would constitute an undue burden, while the undue burden threshold is very high. The way in which organizations provide effective communication was always intended to be flexible. And let's face it, the contents of a website are first and foremost forms of communication.

So if an organization made their goods and services just as readily available through, say, a telephone line as through an inaccessible website, would that provide effective communication? Well, in the few cases that have looked at the issue, the courts have consistently said no.

But I think that's because they had lousy facts. And Thurston versus Midvale Corporation, for instance, the defendant didn't present any evidence that contacting them by telephone wasn't affect was as effective as using their website. And just last year and Domino's Pizza, Mr Robles had to wait 45 minutes on the phone before he could order a pizza.

I think this means that it's still unclear if using a telephone based alternative can remedy the effect of communication violations of an inaccessible website. Maybe someday in the near future, a clever company will provide fantastic after hours support for customers with disabilities trying to access the websites of companies that can't or are reluctant to make their websites fully WCAG compliant. In any event, the second element of a good accessibility statement is an alternate alternate means of obtaining the same goods and services that are available on a website. This could be a top, a phone number that is preferably toll free, and that is operators who can help customers with disabilities, including those calling over to relay service. Browse the company's offerings, place orders and perform the same tasks that are available through the website. The last essential element of an accessibility statement is a feedback mechanism. Some users may encounter inaccessible content, but not in a way that triggers discrimination.

For instance, they may see a product on an inaccessible page that they are curious about, but they're not necessarily interested in buying. This is content that has slipped between the cracks, and users with disabilities should have a means of reporting it so it can get fixed.

Now, what's your feedback mechanism? Looks like it's really up to you. It could be a simple email address, or it could be more elaborate, or it could be more elaborate form that asks for specific information like the URL of the content, why it's inaccessible contact information, etc. I tend to like asking for more specific information because users almost always provide scant information that can't be used effectively. Plus, if they provide contact information, an organization can show its appreciation by giving group discount coupons for other rewards to help bolster customer support. After all, satisfied customers are less likely to file complaints.

The California Online Privacy Protection Act, or Cal OPA, requires that companies post their privacy policies on their on their homepage and on any page that requests user information to keep things simple. I think that posting your accessibility policy at the same location is the easiest thing to do, but certainly posted it on your homepage is important at a minimum. OK, so that's the easy part. Developing a logical accessibility statement and posting it literally takes only a few hours and you're done. Stick to that accessibility. Stay sticking to that accessibility statement. It's harder. But if your company has great customer service, that shouldn't be a problem either.

And if your organization has lousy customer service, it's likely that accessibility is not its biggest problem. The hard part is actually baseline in. And achieving the rest of Web accessibility here. There are three strategies that need to be implemented.

The three, these three components are automated testing, manual audits and lifecycle improvement. So let's talk about these steps from the simplest to the hardest. When people first start their journey to make their websites accessible, the first thing that they instinctively turn to are automated solutions.

Web pages are computer code. So there must be a way for a computer to analyze and fix it, right? Well, if web pages were written to be understood by computers that may work, some web pages are meant to be read by humans, and that means that it ultimately has to be understandable why humans take the most basic and universally understood work requirement WCAG one point 1.1 requires that images have all alternative text that serves an equivalent purpose. We think of this requirement mostly in the context of making on screen images accessible to screen reader users.

While an automated testing tool can test to see if an image has an art or area hyphen label attribute, it can't identify the image. Even if the tool used highly advanced artificial intelligence to identify the image, it couldn't identify the author's purpose in using that image or whether that purpose is adequately described, and it is identifying the purpose of the image that is standard in WCAG one point 1.1 and missing alt text for images is an area where automated testing performs better than most other areas of work. The fact is that most of the WikiLeaks success criteria are written with deliberate focus on whether the human experience is faithfully conveyed to users with disabilities, and computers are pretty terrible at understanding what that human experience is. In fact, if you look at the 38 level and success criteria and we take 2.0 or the 50 level, a success criterion which 80.1 only about 25 to 30% can be accurately tested using automated testing, and that includes success criteria like we take one point 1.1 looked at through a much stringent stringent lens. However, there are there's only one success criteria that can be accurate, accurately tested with automated scanning, and that's wicked four point 1.1. In fact, for this success criteria, the only practical way is to test it with automated testing tools.

If automated testing is so limited, then why do I recommend it? The reason is that even its limited view can uncover a lot. For instance, automated testing tools can test with 100% accuracy if an image tag has no all to attribute or ari hyphen label attribute at all.

It can also test if a table has no head or cells and hat and is not identified as layout table, and these basic accessibility problems happen constantly. So having a tool that can rapidly scan entire websites is vital.

In short, automatic testing tools aren't useful because they're perfect. Instead, they're useful because humans are so imperfect. Plus, because most automated testing tools are great at processing large volumes of data, they also give managers useful metrics about an organization's general success or failure over time and achieving accessibility.

While those metrics are far from perfect in gear and can still leave an organization open to risk. There is still a useful objective measure of accessibility trends because automated testing tools are so limited. The only way to really understand whether a web page.

Note not a web site is accessible is to manually test that web page. Manual testing is very labor intensive and typically takes skilled tester anywhere from half an hour to over an hour and a half to test a page.

This level of effort is required because quality testing needs to include the different disabilities that can be impacted by a webpage and a different assistive technologies that users can use. Plus, with more and more mobile users, manual audits also need to include careful mobile testing across multiple multiple devices and mobile operating systems.

With quality manual testing, however, the real bulk of the time comes in documenting the accessibility issues. For most access competent accessibility consultants, this means taking screenshots of the problems identify any associated code and testing alternative code to see if these changes correct the problem.

Then. Good accessibility audit summarizes all this information in a way that empowers developers to examine their code for similar problems and correct them. Obviously, this level of effort can't be expected for every web page, every every page on a website, and each error cannot be documented to this level of detail to prevent audit trails from spiraling out of control in terms of costs. Most consultants review sites with their customers and try to narrow down a subset of up to 25 individual pages. That includes representative examples of all the types of content on a website. This kind of feedback that became this kind of feedback that audits provide is great for developers and gives them exactly what they need to know to fix their websites. The audit report gives them the tools they need to know exactly what they should look for in a website and then go fix it. Now, apart from the high cost of manual audits, there are a few other problems with manual audits.

First, they only identify and fix representative examples of accessibility problems. There is no guarantee that testing the testers and developers will actually find every similar problem or fix the ones that they find. second, unless the organization has a mechanism to track all of these problems, management can't use manual audits as an easy way to track the success of their accessibility efforts. There's a problem, however, that comes up when organizations rely only on manual and automated testing. New accessibility problems keep, keep, keep getting introduced into the website. Now, the reason this happens becomes clearer when we look at the typical web development process.

In most web development projects, creative teams start the process. This includes UX designers who layout the overall who layout the overall layout and design of a website. It also includes copy authors and graphical artists, and the problem is that manual audits and automated testing results rarely reach the creative teams.

Instead, developers and Q8 teams get stuck with the job of identifying and fixing the accessibility problems. While the creative teams continue to create inaccessible designs. Now this is a lot like using a bucket to bailout a boat when there's a huge hole hole in the hull.

As long as problems keep entering the system, no amount of catching up can solve the problem. Obviously, in order to fix the problem, the solution is to bring creative teams into the solution. It isn't hard to train the UX designers how to annotate wireframes so developers know how to code web pages.

And it's also easy, for instance, to try and copy authors how to write alternative text or to train graphical artists how to meet non-tax contrast requirements. While that seems relatively easy, fully developing a program for companies so that creative development and teams understand work and support each other is usually a tall order.

The first part of the problem is really a people problem. Accessibility always requires a strong commitment from leadership. This isn't necessary because accessibility needs to be expensive, but accessibility does require some degree of change, and change always feels dangerous and uncomfortable.

Accessibility initiatives also require buy in from the right people. This obviously depends on the focus of the initiative. Fixing a bad procurement process or making sure that maintenance projects in a built environment are done correctly obviously requires an entirely different casts of characters.

For web accessibility, however, it usually comes down to the creative development and key team leaders. I can't tell you how to master the soft people skills to make this happen, and it's beyond the scope of this presentation to tell you when tools like focus groups or pilot projects make the most sense.

Those are topics that people like Kevin McDaniels are much better equipped at talking about. But I can't tell you that one tool that is absolutely needed for baseline and implementing web accessibility project is a clear set of accessibility resources, which I'd like to talk about next.

Remember those problems of work that I talked about earlier? For instance, the fact that the WikiLeaks success criteria are vague, and so it's often impossible for developers and designers to know exactly what to do. The first thing that needs to be done is developing a clear and objective understanding of exactly what web accessibility means in the first place. For instance, we can go one point 3.1 is vague about whether area landmarks are required. So is your organization or required? So is your organization going to require them or not? If you go through the exercise of reviewing the 50 WikiLeaks success criteria at Levels A and you'll end up with about 80 to 90 specific and objective core concepts that fully cover with CAG, knowing that your organization wants to implement ARIA landmarks is one thing. Making sure everyone knows exactly what they have to do is another. All too often, we just assume that developers will take care of accessibility problems unless creative teams no exactly how to design for accessibility.

However, developers are often left guessing and end up making inaccessible content. We can't. It's all about making sure that the user experience for people with disabilities is comparable to users without disabilities, and that user experience is exactly what UX stands for when it comes to UX designers and engineers on creative teams.

This means that every core concept needs to be broken down by role. For instance, if an organization is serious about Aria landmarks, it needs rules specific rules that tells UX designers that they need to annotate their wire frames, identify the landmarks so that developers can code them correctly.

And while they're at it, we also need. We also need rules specific rules for Kiwi teams. So why is that? Well, remember what I was saying earlier about all the work that goes into a manual audit? That's because testing a web page thoroughly is very labor intensive and usually requires knowing how to use difficult programs like screenwriter's. As most of us know, screening or is screen readers are just about impossible to use unless you use them constantly. Fortunately, there are about a dozen or so freely available tools on the internet that can tell users what information would have been available to screen readers.

The problem is that none of these tools cover all of the KEGG perfectly, so we need clear rules that tell Kiwi testers exactly which tools to use and when. Once you have requirements at the granular level like this, or I'm sorry, once you have requirements at this granular a level of detail, it's easy to create separate checklists for different teams that describe exactly what needs to be done. For instance, you can create a separate checklist items that state that UX designers must annotate their wireframes, so identify different areas. Landmarks that developers must implement those landmarks using both the proper HDR or five element and the corresponding roll attribute and AQI teams need to test their webpage using the right accessibility bookmarklet in Chrome and match the regions shown by the bookmarklet with the UX designers wireframe. The real beauty of this approach is that creates dependencies into web development process and accountability between team members.

For instance, if a web developer gets a set of wireframes that don't include the proper ARIA landmark annotations, they can't check their corresponding requirement off their checklist. So they need to go back to the wireframe designer and ask that they provide this important accessibility information.

Plus, knowing that knowing the exact steps that a QR tester is going to take when reviewing her designs, a developer will likely take extra care to make sure she met her end of the deal, including the web page.

Otherwise, she knows that the tester will just send a design right back to her later. So creating this kind of compliance tool obviously requires a lot of subject matter, expertize and accessibility. Ideally, you'll also want to create a set of training videos.

Team members can learn on the job, but the real headache and this kind of effort is the amount of time needed to maintain it. The largest I.T. companies like Microsoft and HP spend a lot of resources each year to make this happen.

But like the Papa John's Pizza commercial, this is where I say, and then you realize we've already done this for you. My business partner, Jeff Singleton, and I realize that most companies have neither the expertize nor the financial resources to create this kind of compliance tool.

So we created Web, a line that breaks with cake, with cake and down in exactly this kind of way. Each of our core concepts are broken down by role, supported by about 90 training videos that tell creative development and Q8 teams just what they need to know so they can get it so they can apply it and get back to work. We also created all the necessary checklists for copy authors, UX designers, graphics and media professionals, developers and QA testers. And while it's not quite as easy to implement as simply popping a pizza in an oven, Weber line gives ambitious program managers exactly the tools they need to succeed.

Over the last year, it's been fascinating watching how different organizations use these kinds of tools. Some organizations just swallow the bitter pill and implement work all at once. That's great for companies that work on specific, web based applications that they're trying to sell to government buyers and need to create an accessibility conformance report.

Other organizations take on chunks of issues at a time. This works great for large organizations with lots of developers and a ton of web content. They may dedicate a month to only fixing their online forums, and so they cherry pick the specific rules for forums which work inconveniently scatters everywhere and only focus on those before moving onto a different topic the next month and other organizations that have relatively that have a relatively good understanding of accessibility. Simply use tools like web online as knowledge basis to explain a particular accessibility concept, while also showing how it fits into the overall structure of WCAG.

So exactly how an organization uses knowledge and compliance tools like Web of Life really depends on the needs of the organization. Deciding which way to go often comes back to the soft personal skills of good program management. one advantage, however, that all users have noticed to having crystal clear requirements is that it makes it much easier to track success over time. After all, if you have a checklist of clear requirements for each role of a web development process, you know exactly how work is being met. Even before a single line of HTML is even written, you'll know your forms are labeled.

Your images have the right text and that your wicket, and that all the other work requirements have been contemplated at a design stage. So all the developers have to do is write the code. But I think that the most important outcome of combining the personal skills of a great program manager with the knowledge tools like where the line is, that it creates a self-perpetuating culture of accessibility team members and not pesky managers. Follow up with each other to make sure that accessibility gets done. People start to look positively towards accessibility instead of dreading it. I feel really blessed to have worked both as a lawyer and a consultant in web accessibility.

I've had the opportunity at both ends of the field and may very well return the lawyer to the lawyering side in a not so distant future. I think it's important for both lawyers and their clients to understand what goes into baseline in an implementing web accessibility.

It's not impossibly difficult, but it's also not as simple as just adding a few lines of JavaScript on your web page and letting your users fire up an overlay. And it's also not as simple as just buying an automated testing tool instead.

Accessibility is about human beings, and that means that the solution is more difficult. It requires people skills and getting people excited, knowing that they're helping other people while they're much more challenging. This kind of problem can also be much more satisfying in the end.

- END OF TRANSCRIPT –