

**SOFIA LEIVA:** Thanks for joining the webinar entitled "Integrating Accessibility into the Development Process." I'm Sofia Leiva from 3Play Media, and I'll be moderating today. And today, I'm joined by Monica, accessibility technology lead, and Michele, accessibility program manager from Oracle. But with that, I am going to hand it off to Michele and Monica, who have a wonderful presentation prepared for you.

**MONICA GAINES:** As Sofia said, this is Integrating Accessibility into the Development Process. Michele, Kent, and I actually presented this at CSUN this past March. Michele is our accessibility program manager at the Accessibility Program Office, and I am the accessibility technical lead. And just for completion, even though Kent is not joining us today, he is our accessibility program director. The next slide, Michele.

Thank you. So I need to get through a little legal logistics. The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

So a quick peek at our agenda, I'll be going through accessibility in the development process and creating accessible product, and Michele will work with our resources, accessibility by role, and then we'll have some time at the end for, hopefully, some questions.

So where does accessibility fit in the process? These are recommendations for how we would fit them in into your company's development process. You should choose those steps that work best for your team.

So we'll start with you should actually define an accessibility policy for your organization to follow. This will allow you to make sure that you get approval from your top executive, legal, management, government affairs, et cetera. You'll want to have guidelines or checklists to enforce, for example, are you following with WCAG 2.1 A/AA.

Also, document and include your testing procedures. How do you expect your teams to go and test for accessibility? Document some development best practices. Make sure that you have a bug tracking system, where you can actually track your accessibility bugs. You'll want to

provide some training for your groups.

Communication is key, how you're going to communicate accessibility information, updates, et cetera, to your development teams or your entire organization. And then, also, we use the Voluntary Product Accessibility Template, or Accessibility Conformance Report, to document our level of accessibility for our products. And you'll want to probably have some kind of approval process for the writing of those in your team.

Next slide.

So your policy should include applicability and scope. What does it apply to, all products and services. Make sure that you include authoring tools, any hardware that you might sell, documentation, training, any developer tools, et cetera. Also, how are you going to enforce these things? You need to have something for the authority enforcement, and what you're going to do if there's violations. How are you going to track? What are your checks and balances? How do you define success and failure?

And then finally, we found that it's a good idea to include some kind of grace period.

Accessibility does not happen instantly and teams will need to build it in, specifically if they're having to do it for existing products that don't have accessibility. So they may need to define some remediation roadmaps.

So this is just kind of an image of what you might be going through for a development process. You'll want to start at your planning, all the way through to your release. And so where does accessibility fit into that? And then, what type of development model are you potentially using? Is it waterfall? Is it iterative? Is it continuous?

So there's no single way to integrate accessibility into the process because each product is unique. You'll need to assess what works best. But it's ideal if you can actually include accessibility into each phase of your process, starting with planning. It should be in your designs, your UX designs.

Obviously, development will need to code and do some unit type testing. Documentation is also going to need to make sure that they're including accessibility in your online documentation. Both of those will fold into you need to test for them to make sure that accessibility is working and this has been implemented correctly. And then finally, how you're going to document that in your VPAT. And then finally, you'll release your product to your

customers.

So the way that you do this would depend on your development model. And I'll go on to a bit more detail, spending some time on waterfall, which can take years, and continuous, which would be done in hours. And I think I'll spend more of my time on the iterative, which can happen in a matter of weeks.

Next slide.

So before we look specifically at each model, there are some general things that you want to keep in mind with respect to integrating accessibility into your process. You'll want to add accessibility into your planning and design phases. Your UX design team is already using personas when they go and look at how I'm going to do my workflows. You should also consider doing user stories or interviewing people with person with disabilities, including those types of use cases.

It's also really good to include accessibility into each of your use cases. This way, you will have accessibility built into each feature slash use case, as opposed to having a single use case at the end that says incorporate accessibility, which is very easily cut. Also, include accessibility in your implementation and verification. You'll want to make platform and architecture decisions that include accessibility.

Make sure that accessibility is part of your unit testing. And if possible, you'll want to include testing by a person with disabilities, as they will give you a very unique, specific way that they're going to use the product, as opposed to someone who does not have a disability.

So your waterfall development model is probably something that's mostly used for hardware or operating systems at the moment. Your process flows in one direction, from planning to release. It can take several years to implement. And you definitely want to include it into each phase. So in your planning phase, you'll want to make sure that you include your accessibility requirements.

Design, you're including it in the actual design. When you go and implement it, your accessibility features and capabilities are part of your code deliverables. Make sure that you're testing it as part of unit function and system testing. And then finally, prior to your release, make sure that your accessibility features are highlighted in the documentation.

Next slide.

In an iterative development model, more commonly known, I think-- there's probably many out there, but I think the most common one is Agile-- this is more of an adaptive approach. There is feature-driven development, and the team will adapt to changing products dynamically. Development process completes in weeks. You will likely have multiple iterations which would be required to release your product. The product is tested very frequently through these iterations, which can then minimize the risk of failure in the future.

In Agile, your process is broken into small, incremental builds or sprints. Within each sprint, your product features are going to be described by stories in the voice of the target user. You will identify the tasks to be implemented and tested for each sprint. And at the end of the sprint, features implemented, and it should work completely as intended.

So you're having working software for your stakeholders to review. And then at the end of one of those, you may shift. You can integrate accessibility into one of two ways. I'll go into a little bit more detail. One is dedicated accessibility sprints or include accessibility in each sprint.

So, some advantages or disadvantages to having dedicated accessibility sprints. Your advantages, you can focus completely on accessibility. It might save you some implementation time over the long term. It supports the notion of having an accessibility specialist for development and test. Perhaps you'll have someone else come in and consult on those features. And then it supports a focused system testing only for accessibility.

Disadvantages are that, if you add accessibility in a later sprint, you will likely involve rework, which will end up taking you more time. Accessibility issues found later could be harder to fix. It's possible that your design could be locked by the time your accessibility is implemented, which, again, would make it harder to change. And then, there's no undoing of previous sprints.

Next slide.

Next slide, Michele. Thanks.

Some advantages or disadvantages to having accessibility in each sprint. Advantages are you will have useful features at the end of each cycle, which will include accessibility. It's also more reasonable that stakeholder feedback would include your accessibility. Supports a rapid, incremental, and continuous development model.

Supports continuous, consistent development and test resources. And you can catch and eliminate your accessibility problems much sooner. The disadvantage is the accessibility knowledge can be rare among an entire team, and so you would probably require an expert to provide guidance on accessibility.

**MICHELE VAN DOOZER:** One thing we want to flag on this is that we stole this information from the IBM [INAUDIBLE] presentation. So we want to give credit where credit's due on it.

**MONICA GAINES:** Yes. Thank you, Michele.

And then, from a continuous development model, it is similar to doing iterative, except you are actually pushing to production more frequently. And it's possible that you can have multi releases in a single day. So you have a very much accelerated process. Teams would have to generally automate as much as possible. And done means that your working code is deployed. And you will need to integrate accessibility into each and every chunk of the deployment and development.

And I should note here that, when I'm talking about automation, you also need to take into consideration that only about 17% of accessibility standards can be automated. Human intervention is, in many cases, required because you need to interpret the results. So it does factor into that process. And then I'll talk a little bit about creating an accessible product.

So if you're creating a brand-new product, you will want to design, build, and test all of your accessibility features from the beginning. It's much easier to do this than it is to go back and retrofit accessibility at the end. It's helpful if you identify the type of product that you're building. Is it software or hardware? If you're software, am I looking at a command line interface? Am I web? Am I doing mobile? Am I doing many of these, all of these?

Because you may actually be able to eliminate or not have to worry about some of the features or some of the standards that you've documented that you need to be following based on the type of software that you're building. Make sure that you're referring to your appropriate checklists. And when I'm talking about checklists, we actually have some examples of that in a moment for the different types of UI that you're using. And make sure that your documentation is also coded and tested for accessibility.

For an existing product, it's incredibly helpful if you go and evaluate your product for its current level of accessibility. If you're adding any new features, make sure those are going to be

coded, designed, and tested for accessibility from the beginning. For the existing features, you want to prioritize those for how you're going to incorporate or integrate accessibility. Start with your core product tasks and most commonly used features, and also make sure that you consider complex or unusual features, as they may require more redesign and additional work. And--

**MICHELE VAN** OK, so we're going to--

**DOOZER:**

**MONICA GAINES:** --sorry.

**MICHELE VAN** --we're going--

**DOOZER:**

**MONICA GAINES:** Go ahead.

**MICHELE VAN** --to move to my part, the resources. And my apologies, evidently when I hit page down it takes

**DOOZER:** quite a while for you guys to get it. So I might get ahead of my slides. So, some things to start with on some resources. You need to figure out what standards you guys are going to be using. The highly recommended ones are WCAG 2.1 and Revised Section 508.

But the biggest thing from your standpoint is figure out which standard you want to follow and make sure it meets all of your customer needs. If you follow WCAG 2.1, you're going to meet a lot of the European requirements and the Canadian requirements. But be aware Revised Section 508 only calls for WCAG 2.0. 508 will not be updated to pick up the new WCAG. But I'm sure your customers won't mind if you meet a newer standard. That won't be a problem.

Monica talked about checklists. And what we're talking about there is helping your teams understand what the criteria is they have to meet. And instead of telling them they need to meet WCAG and 508 and the EN 301549, it's just easier to make one checklist for all of them to follow, or I should say one checklist per type of UI.

So what we've done at Oracle is we have different checklists based on the technologies and the user interface. So we have a web software one, then we have one for the non-web, so your command line, your Java apps, that kind of thing. And then we have also for mobile, and then another one for documentation and support services.

Now, the reason documentation is there is, if you look at Revised Section 508, not only does

the documentation have to be accessible and meet the WCAG criteria, there's additional requirements that we have a checklist or doc to flag that. And then, of course, there's separate checklists for your hardware. So again, make it simple for everybody to follow and you'll probably have a better success rate.

So when we talk about checklists, what we have in ours is we have the guidelines, the actual standards, so that nobody's trying to interpretate and put their spin on it. It's just a flat-out standard. People can read it. But there's also understanding links.

You know, a lot of these standards, they have information for how to implement it, what to do, you know, examples what's right, what's wrong. Use the information that's there. Don't create your own. So we have links in our checklist to those.

We also have sample remarks for when authoring the VPAT so that people can clearly document in the VPAT how we know we've met that criteria. So we give it as an example to [INAUDIBLE]. And then we also have testing techniques. Not only will this checklist explain what the people need to do and what to look for, but there's also, when you test it, how do you actually test to make sure you've met the criteria.

We didn't touch on bug tracking. Bug tracking is really important. And I know every team at Oracle kind of does it a little different, so what we do is we make them tag bugs and tell us how they tag them. So it can either be with an ACC, A11Y, 508, whatever. Each team can do their own thing, but they need to be able to tell you how to find all of their bugs.

When you look at the bugs, assess things just as you would for any other non-accessibility issue. So I've had people tell me that, oh, the fact that the login screen doesn't work with a screen reader, that doesn't impact too many people, we don't have to really make it a priority. Wrong answer. If it's that I can't log in period, that's a problem. That would be a P1 issue. And the fact that this happened only with a screen reader user is a non-issue, that shouldn't make any difference. So evaluate things regardless of the assistive technology being involved.

Also, document your open accessibility bugs. I highly recommend this be done in your VPAT. And that way, any customer knows what they're getting into with your product well in advance. So bugs, everybody got them. Admit it. Put it into your databases or into your tracking system, and then put it into your VPAT.

So, bugs. How do we write them? Don't do the good old fix accessibility. You know, that's

pretty useless. How do you know when you've got it fixed? You need to make a balance between one bug per issue and to the point of you've got one bug for many issues. If you wrote one bug for each alt text that was missing off your images, you'd probably spend more time writing the bugs than fixing the code. The goal is fix the code, and then you don't have to do it, but get a balance there.

And, you know, from the standpoint the product team is going to have to decide what's the right balance. I'm in the Accessibility Program Office. I can't tell you what the right balance is because I don't know your products. But you know, make it flexible in your policies, but make your bugs reasonable. One bug per screen, one bug per criteria. Those are hard to know when you're done because you could have one screen that has 15 issues on it, and you fix 14 of them but technically you still can't close the bug. That's a bad way to do it. So find a balance.

The other thing is consider publishing bugs so that the customers can get access to it. We put our bug information on a short summary plus the bug ID in our VPAT so that customers can go to My Oracle Support and get information about the bug. When you're writing bugs, be sure to include information on how to reproduce it, but also document any workarounds. That way people looking at things can see if they can work around it or if it's really going to be a showstopper for them.

Another resource is testing tools. There are lots of tools out there. Use them. Don't reinvent the wheel. You know, within Oracle, we have the ADF tool kit, which has the JDeveloper audit tools. If you're using our products, use our tool. If you're using an HTML user interface, there are a whole bunch of tools out there available.

These slides will be posted and the links will give you access to other pages and pages of different tools that are out there. Color contrast checkers, you know, there's a bunch out there. You know, I looked at the Paciello Group here.

There's also add-in tools for different browsers. So Chrome, Firefox, they each have them. They're from different agencies. Some are free, but most of them-- excuse me, most of them are free, some of them you have to pay for, but be sure to use the tools. As Monica said, they test generally somewhere between 15 to-- the statistic I hear most often is it will test 17% of the criteria. Remember, you have to do your manual testing, but use the tools. They are there.

Another tool to think about is your release management tools. Oftentimes, you will have a

process that you release your products through. And there will be tools to track things such as your export control, your security issues. Well, why not add accessibility to that?

At Oracle, what we do is we have the requirement of you have to have a VPAT before your product can release. So the tools themselves map to each other. We don't expect anybody to do this manually because, if you do it manually, it's not going to follow through. So get your tools to do it.

Our tools are smart enough that the VPAT, you can map your release in your product lifecycle or release management tools, you can map to your VPAT well in advance of your release, but the tools are smart enough that, if the VPAT is not in an approved or published state, the product can't release. So we use that as a gating item. So highly recommend look into your release management tools and see how you can tie accessibility into that.

So, accessibility by role. As I mentioned, Monica and I are in the Accessibility Program Office. We refer to that as the APO. What we do is we determine what standards we want local products to follow. And that way, it's our job to manage or monitor worldwide what's going on with regards to accessibility requirements, and make sure that our checklist for the groups accommodate all of those different criteria.

We push for harmonization of that criteria, but it doesn't always happen. But that way, we determine what standards we want to follow, we develop the coding and testing guidelines for our teams to follow, and then we do training of our teams. We have an annual accessibility conference at our headquarters that people fly in for, but we also record those so people can watch them at other times in the year.

Then we also give help, guidance, [INAUDIBLE] testing of products. If you ever, as a product team, give me a VPAT that says your product is perfect, oh, that's a nice red flag for us. We're going to go try it out and make sure and see where you are, because I'm sure there's going to be some little snags here and there.

We also review and publish our VPATs and post them out on Oracle.com. And notice I say we review them. The bottom half of this slide says we don't write them. The product teams are the experts about the products. It's their responsibility to write them. We just review, make sure they make sense, and question as needed. And we don't make the products accessible. We teach the teams, but it's up to-- they're responsible to make it accessible. We'll just guide them along the way.

So accessibility applies to everybody in the products area. Service, support, all of that. Everybody has to be aware of accessibility and how important it is. That's where Monica had mentioned it's really good if you get an accessibility policy that your company has to follow. At Oracle, we have it. And if teams go, oh, we don't have time for accessibility, I can just pull out the, well, this is on the compliance and ethics page, you need to do it, and all of a sudden people start taking it seriously.

Also make sure that you understand the accessibility guidelines that you're going to use, such as WCAG or Section 508, and make sure-- this is a big one-- make sure people get accessibility training. Some roles require more training than others, but there will be training for everybody involved. There was a lawsuit-- I think it was last year-- GNC got sued by a blind person that wasn't able to use a screen reader on the GNC website.

I bring this up because it goes back to, you know, in the court of law, there's three criteria that you really need to be careful with. You know, is the person an expert and qualified to be doing the role when it comes to being a subject matter expert with regards to accessibility? Did they use a reliable method for the process they validated the product with? And do they have a report that, in the case of a lawsuit, can show the judge how you came up with the information you did?

In the GNC lawsuit, they had these awesome reports, but the problem was, one, the person that did all the accessibility testing didn't have training in accessibility, and two, he ran their website through two online accessibility checkers, which, as we noted already, as you already know, covers 17% of the criteria. So if you do just automated tool testing, you're not meeting the requirements. So in this case, GNC, they got in a lot of trouble. Don't go there. Train all your people, make sure you have processes to follow, and then you'll get some good results.

So program and product managers, what kind of training do they need? For the program manager, they need to track the accessibility progress of the entire product. You know, make sure things are going forward. You don't want regressions in your releases as the product goes along. Ensure engineering is implementing features and doing their unit testing.

Notice that it says engineers and unit testing. It's not just QA that tests. Has to be upstream everywhere. And program managers need to ensure that happens. They also ensure QA knows the criteria and what they're going to be testing for, and then helps coordinate getting

the VPAT done. We'll talk about VPATs a little bit more, about who should be doing what there, but the program manager needs to be involved.

So the product manager, they're responsible for making sure the requirements are there in the release so that it's not something that we think about and add at the very end. They're also responsible for working with the product teams to evaluate, prioritize, integrate accessibility into-- excuse me-- integration of standard into the product.

User experience, these guys are key. Of course, you're going to hear me say that in just about every role, but if the UI is not designed from the beginning to be accessible, you're going to probably fail. So it's critical that the UI designers know what accessibility is and know how they need to incorporate it into their design.

So, you know, at Oracle, we have a special checklist for the UX designers, making sure to remind them things. You know, how everything needs to be done from the keyboard for each image, how to get the alt text as meaningful. When they're picking colors and stuff, they need to make sure they have the right contrast ratios and reading order. So those things are critical, and it has to be planned from the very beginning.

Engineering. As noted in the GNC lawsuit, make sure they're trained. Make sure they know what they're doing, what they're coding for, and their test methodologies to make sure they've got it right. They need to develop the products using the best practices and guidelines. Make sure that they're doing the right thing, that they understand the tech stacks that they're using and how it addresses accessibility.

While you can have a tool kit that has individual components that are accessible, you might find a way to put those multiple components together, somehow accessibility gets broken. You need to ensure it all works together.

And last part-- I kind of hinted to this before-- they're responsible for the first part of the testing of the UI. You can run automated tests if they're available. Awesome, use them. Build unit tests if you can. But make sure that the first level of testing is being done at the engineering and the development stage.

QA. This is where at the end, or even-- depending on what cycle you're doing, they are critical for testing. So make sure they have the training also on the methodologies needed for this. They are responsible for using recommended testing procedures and identify accessibility

defects. Write good reports so that people understand what the problem is and how they can repeat it and that kind of thing. And then build regression tests if you can. That way, things are improving with each release and able to be caught faster.

Docs team. Again, you have to train your doc writers. They need to understand about the testing methodology for their documentation. So they provide and test their documentation, but also-- the last bullet there is important-- you have to document the accessibility features of the product.

The assumption that people will know what little switches to set for supporting accessibility if you have them, don't make that assumption. Document. And hopefully your making standard UI accessible for everybody and there's no special accessibility modes. Please make those go away if you have them, but the accessibility of the product has to be documented. Excuse me.

So VPAT authoring. This is important. At Oracle, what we've done is we have two different roles involved in authoring a VPAT. The first person is the author. They write the VPAT. So they gather the accessibility testing information, and we don't even start the VPATs unless the testing has been done. They gather that information, they create the VPAT based on that testing information, and then they're responsible for keeping that VPAT up to date.

So with every release, you know, it'd be nice if we could do VPATs every single release, but most teams can't do that, especially when you have a release cycle that's every week. It's just not possible to do VPATs that often. But the VPAT author needs to ensure that that VPAT is accurate for every one of those releases, and then issuing a new one as necessary.

To keep things in track, what we've done at Oracle is we have a VPAT reviewer, which is a different person from the author. So this person is knowledgeable about the product, but also about accessibility. And they can review the VPAT to make sure that the information's correct. Excuse me. You'll notice we don't say anything about marketing here because we've found that marketing people write awesome VPATs that have nothing to do with the product.

So make sure that you're involving the right people, the people that are involved engineering-wise, QA-wise, they know the product, they know accessibility. And that way, you have more accurate VPATs. So the VPAT reviewer, they're one of the last people in the product team that they read the VPAT and they approve it before it comes to us.

What happens, what I mean by us is, at Oracle, the author writes the VPAT, the VPAT

reviewer reads it and approves it, fixes it as necessary, and then it comes to the APO, Accessibility Program Office. We do a review, and if we approve it, then it goes to the product line VP for their approval. And then it will finally get published. So we have some checks and balances and make people accountable for things within the product line.

So don't forget your training and sales organizations. These are customer-facing people that need to know about accessibility also. So the training stuff, you want to make sure any in-person, online, all of that training, it needs to be accessible, including the videos. You've got to have your captions for the hearing criteria, and also your described audio for the vision criteria. Make sure that those are included.

And sales, you need to ensure that sales people know what VPATs are and how to use them, and make sure they apply them to the customers. And often, we have the sales team involve the APO as appropriate for product demos. You never want your sales person to think that they can run a screen reader and show the accessibility of a product. That's not cool. Our situation, what we do is we have the APO involved with that. As you all know, screen readers are hard to use. So make sure that people that know how to use assistive technologies are the ones doing the demo.

So to wrap it up-- we're getting through this conference. So be sure to create a corporate accessibility policy. You want to incorporate all phases of development in the process. Ensure your teams understand what accessibility is and how it impacts them, and make sure they're trained on the standards that you're going to be following and how the tests work.

Utilize your checklist, tools, and resources. Start testing early. Early, often, always. Every release, if possible. And then document the accessibility of the product. Once you have your test results, write your VPAT based on those test results. If you don't have a test result, don't write a VPAT, because you really don't know what to be including in there.

And then, accessibility is not something to do once and forget. You guys have to do this every single release. You want to make sure you're always going forward with this. You don't want to be going backwards. We hear often about complaints of-- or we used to in the past, hopefully less now, but it used to work. You have a new release, now it doesn't. That's the worst thing you can do to your customers. So make sure that you do this and do it with every release.

So I guess at this point we want to open it up for Q&A.

**SOFIA LEIVA:** Thanks so much, Michele and Monica. I'd like to encourage everyone to keep typing your questions in the Q&A window and in the chat window, and we'll add them and hopefully get to them. The first question that we have is, do you have favorite tools you use when designing or developing new accessible products?

**MONICA GAINES:** So from a UX perspective, I know there are many different tools out there. And unfortunately, most of them don't really output accessible information. I think that you need to do a few things. I've seen some annotations where, OK, this screen does this, and here's features that you might need or attributes you might need to set for accessibility.

There are some tools that-- there are some plugins, I think, for [INAUDIBLE] specifically that has a color contrast tool on it so that you can go and do some accessibility testing on your designs itself. But it definitely, the onus is to make sure that you are documenting. And don't forget keyboard. That's really key in your designs themselves.

**SOFIA LEIVA:** Thank you. The next question we have is, do you test products with users who have a disability, or how does your testing process look like?

**MONICA GAINES:** If you can, yes, you definitely should use a user with disabilities as one of your testers. I think a screen reader user is the person who knows and understands. That's the tool they use every day. So they're going to find things that someone like me, who doesn't use it, who only use it for verification purposes, could possibly miss. So if you have the ability to use them, you should.

**MICHELE VAN DOOZER:** But it also goes back to, you know, if you-- everybody should-- it should be mandatory that some sort of assistive technology is used during testing. But yeah, if it can be done by a person with a disability that's familiar with the tool, that is the best because that's where the rubber meets the road.

**SOFIA LEIVA:** Thank you. The next question we had was, what were your operational considerations for deciding on your current workflow?

**MICHELE VAN DOOZER:** I'll take that one. Most of it is looking at what the different product teams have already done, because if you go in and try to get them to just, you know, forklift change everything, you're not going to get buy-in. So your biggest thing is to go to your teams and find out what their workflow is, and work with them to make sure accessibility is added in the appropriate places.

**SOFIA LEIVA:** Great. Thank you. The next question we have is, are there any specific compliant points

depending on the region or country?

**MICHELE VAN DOOZER:** That's where we're trying-- Oracle is a member of ITI, which is the Information and Technology Industry Committee. I think that's what the C is.

It's big that we try to get harmonization worldwide. That way, we don't have to have different rules for different countries. So if you are members of different organizations-- there's different ones worldwide working on trying to be sure there is harmonization-- get involved and make sure that we can get that on track. The biggest thing is, by having one standard that we can use worldwide, then that makes it best for everybody.

**SOFIA LEIVA:** Great. Thank you. The next question we had is, in a world where you have no team, what can you still accomplish, and how?

**MICHELE VAN DOOZER:** Oh, my. You have no team. I'm assuming that means you have no APO. And the best thing is to do a grassroots within the product teams.

You know, just do-- if nobody-- if, say, you have a scenario of you have nobody in your group that's a person with a disability, learn how to use some assistive technologies and then start doing demos with your teams to show them, hey, are you aware this is how your product works or does not work well with an assistive technology?

Because so often, we'll approach people that are new to accessibility that are like, you're just another requirement, why do I have to do this, I've got all these other, you know, higher priority. But if you start doing a demo and really walking through with them and showing them here's why we're asking for what we are asking for, all of a sudden it's like a light bulb moment, and they realize, well, first of all, it wasn't really that hard to fix what you've asked for. And if you catch them at the beginning, it's easy to add it from the get-go. But a lot of the time, it's just work with the teams to get some empathy going, and you'll get really good results.

Monica, you have anything to add there?

**MONICA GAINES:** No, the demo part is pretty key. We recently were demoing a demo page that we had where we had the users who had never used the screen reader before try and navigate using a keyboard to that particular field, and how they would go listening to it and enter those things. And there were definitely some aha moments for them. This was some UX designers, that they had not actually tried to do it. So they're going to look at how they utilize those

components differently, or how they design them differently based on that information.

**MICHELE VAN  
DOOZER:**

But we even do hands-on jobs classes for some of our people. And the first part of the class is just the plain old can I do everything from the keyboard. And it's quite hilarious because we'll go to someplace like, you know, an airline website and try to get people to just do the most basic. I want to go to this site, I want to use this [INAUDIBLE]. I want to book a flight from-- you know, pick two towns and a date, and how you're going to pay for it, but force the people to do it on their personal computers, so they're aware, but don't use a mouse.

And believe me, you're going to have to stand there and really keep on them about don't touch that mouse. And you know, it might take them a good 10 minutes to just find the flight. And we turn it into a joke going, so how easy was that? Well, let's now try it without the screen. Let's turn on the screen reader and see if you can do it again. And all of a sudden, it really gets people going, well, you know, I couldn't hit this part of the page. Yeah, because it was coded wrong. So all of a sudden, you know-- getting people hands-on really starts to get people involved.

**SOFIA LEIVA:**

Thank you so much. The next question we had is, in your example, it seems the engineering and QA folks might be authoring the VPAT based on test results, but who, what role, usually is the reviewer of the VPAT?

**MICHELE VAN  
DOOZER:**

The reviewer is a person-- it could be anybody within the product, you know, what we call a line of business. But it's a person that understands accessibility, knows the product. You know, that way, they can look at the information and go, well, I know for a fact that the product doesn't work this way. So they're able to answer it.

As to what role that is, it could be a program manager, but it would have to be a program manager that's able to read the test results and understand things. So that's why we purposely keep it vague, but VPATs unfortunately are technical. So you can't have just a program manager writing it from square one. They're going to need help from the development team, from QA.

**SOFIA LEIVA:**

Thank you. The next question we had-- and I think it refers to one of the slides-- is, where does the 17% coverage of requirements data piece come from?

**MICHELE VAN  
DOOZER:**

Actually, that's one of those you can Google it. There's a bunch of different views on how much can be tested via a tool. Because think about it, a tool, yes, it can tell you that you have

alt text on the image, but it can't tell you if that's the right alt text. It can tell you you have headers on the page, but are the text in those headers correct?

So if you say that a tool can-- you know, the fact that it tells you you have a header, well, that counts as you met the criteria, no, that's only partially. So that's where you can never really find a hard and fast number as to automated testing can give you a guaranteed result of x criteria. That's why we said the statistics are anywhere from 15-- I've seen some sites say a tool can verify 30% of the criteria. Not really. That's why Jim Thatcher had originally said 17%, and that's usually the common number we use.

**SOFIA LEIVA:** Thank you. We have a couple more questions. The next one is, what should everyone do on day one when implementing accessibility?

**MICHELE VAN DOOZER:** Work to do on day one. Wow, what's this? Just really, you know, it needs to be included-- people need to be educated. Start at the beginning. You can't do it unless you understand what accessibility is to start with your training and, you know, hopefully get it at the beginning of your product development, back at the UX time.

**MONICA GAINES:** Yeah. I'll add a little bit to that. If you're a developer and you actually have some tools, get the easy items out of the way first. Make sure that all of your fields have associated labels, your images have alt text. I mean, there are some standards that are so easy to implement that, if you get those out of the way from the beginning, you can concentrate on the harder things, like using it with the keyboard and do you have correct focus. So definitely educate, and then, you know, at least from development, make sure you kind of get the low-hanging fruit, so to speak.

**MICHELE VAN DOOZER:** Well also, by getting the low-hanging fruit, you build credibility for accessibility of your product, because if you say I've done all the hard stuff but you've skipped the simple things, people are going to look at it and go, wait a minute, you didn't even do the easy part. Why should I believe you that you did the hard stuff the right way?

**SOFIA LEIVA:** Thank you. The next question we have is, how have you found that accessibility features improve the experience for all users?

**MICHELE VAN DOOZER:** Oh, yeah. Yeah, accessibility, you know, including accessibility from the beginning, it makes it a more usable product for everybody because you're taking the time to think through how stuff works instead of-- you know, the worst case is when you get a bunch of different engineers

putting together their different opinions on stuff and things working inconsistently because you glued together a bunch of different products. That's horrible. But if you think things through and make it accessible and do things consistently, it's improving it for everybody.

**SOFIA LEIVA:** Thank you. We have one more question. What tips or tricks have you learned along the way for getting budget or buy-in for accessibility initiatives?

**MICHELE VAN DOOZER:** The biggest thing is showing, you know, if you can get this as a product differentiator, that could be huge. You know, there are different areas that people are behind on doing accessibility. People, it used to be do I have a VPAT, just stick it in the file. Now I actually want to read it and I want to see you improving your product.

But if you've got a product that is ahead of the game on that, it clearly-- if you're selling to the government and if the agency is following the rules the way they're supposed to, they're supposed to buy the most accessible product, regardless of price. So you know, as I said, it could be a big product differentiator. But if you have to start small with a small budget, go for it and show how this is going to improve things.

**SOFIA LEIVA:** Thank you so much. Thank you, Michele and Monica, for such a wonderful presentation, and thank you, everyone, for joining us today.