Accessible Mobile Apps and Kiosks Transcript

OPENING

Bridget Shapiro
Hi everybody! Thank you for joining Accessibility.com's panel discussion on Mobile Apps and Kiosks. I'm excited to be your host today. My name is Bridget Shapiro. I've worked in operations and client success for over 25 years. Most recently having been the Senior Vice President of Client Management and Operations for just about eight years at the Bureau of Internet Accessibility.

They're a pioneering company in the digital accessibility world. We worked with clients of all sizes from all industries to help them improve their digital accessibility of all of their digital properties. So great book of business over at the Bureau. And today I'm joined by two amazing friends and colleagues, both of whom are thought leaders in the digital accessibility space.

Casey Naiduk and Garry Harstad. So I'm going to pass it over to you guys to introduce yourselves.

Casey?

Casey Naiduk
Okay. Thanks, Bridget. And hello, everybody. My name is Casey Naiduk. It's a pleasure to be here. I am currently the Senior Manager of Digital Accessibility at Bristol-Myers Squibb. Prior to that, I was a Lead Product Regulatory Strategist for Accessibility at Oracle. Prior to that, I was actually one of the founding members here at Accessibility.com leading content experience and accessibility.

And it's my great pleasure to be here with you, Bridget, today. And Garry, who we'll meet in a moment. Some of my favorite people and most respected colleagues in the industry. So thank you and hello.

Bridget Shapiro
Thank you. Thanks, Casey.

Garry?

Garry Harstad
Yeah, my name is Gary Harstad. I've been working with the Bureau of Internet Accessibility for the past ten plus years, working on accessibility strategies, software development and just general a11y solutions all around. And excited to speak about the actual kiosk and mobile

questions that we have coming up today. Hopefully we can spread some light on this very gray and dubious area, which raises a lot of questions.

So let's get to it.

Bridget Shapiro
Agreed. All right. Awesome. Thank you so much, both of you. Happy to be with you today. So basically, our goal here today is to make sure everyone that's in attendance leaves here, leaves the discussion with a better understanding of making mobile apps and kiosks accessible. We want to answer your questions and get to the heart of what you need to know as we all collectively work together to make the world a more equitable place for everyone.

A couple of things just to note This event is being recorded and will be available for viewing later this evening, along with the transcript. You'll receive an email from Lori once that recording is available. Thank you also to our sponsors, the great Bureau of Internet Accessibility and Clusiv for supporting today's event. So thank you both. And I see that there are already a bunch of questions that have already come in, so we're going to try to get to all of them, or at least most of them.

So without further ado, let's just dive in. I'm just going to start going for it.

Casey Naiduk
Let's do this.

PANEL DISCUSSION

Bridget Shapiro
Awesome. So the first question, in no particular order is actually a good place to start. What are the differences between a mobile site, a mobile app, and a kiosk?

Garry Harstad
Oh, nothing and everything. I think is the best answer. You know, when it comes to mobile app development, it's just another app. It’s just mainly targeting a different device. Right? So the principle difference between mobile apps and non mobile apps tends to be screen size, right? Or the viewport available to them. So because you have such a small screen size, you have to make sure that the navigation, the important information is above the fold and that controls and navigation in general is just easily accessible.

There's also a strategy of... for a mobile app, you can be much more concise because you've got so much less space. You're going to be droning on and they have to scroll down to get to the important information. So it's a lot of it is about bubbling the important information and controls up into the upper viewport area of a mobile app with supplementary information being down below.

And in general, the difference between a mobile app and a desktop app as a mobile app has to take those small size into consideration. And additionally, another major point is whether or not they got rotated, right, because you can have liquid designs that work well on both portrait or landscape mode, or you could accidentally, you know, be fixing your, you know, navigation to any working landscape.

And when they look at a mobile, it just looks like a mess or is hard to navigate. So the principle difference are sizes and rotatability in general. There are some nuances as well. Maybe Casey can speak to some of those?

Casey Naiduk
Well, those are good. Those are good intro points. And I'm... I'm almost hearing the question a little more literally. So what are the differences between a a mobile website, a mobile app and a kiosk?

The literal answer to those, I guess, is that a mobile website is is how a website renders on a mobile device. Right? And so hopefully people have built, you know, mobile first or responsive websites that will perform well on a mobile device like a smartphone or tablet. A mobile app, on the other hand, would be typically something available in the App Store itself.

Right. But, you know, those little icons that we click the mobile apps program specifically designed to run on that mobile on the mobile device, typically separate or closed off from a website, but not always. And then, sorry if I am interpreting this too literally, but a kiosk is typically a rendering or the existence of one of those things in the physical world.

Right? So a kiosk is a stand alone, sometimes tablet, sometimes a larger piece of equipment that exists in real life, you know? You could see it and touch it. And it's running a a specific program designed to allow somebody to complete a specific task. Right? So it might be ordering a hamburger. It might be checking in at the airport.

Bridget Shapiro
Right.

Casey Naiduk
Could be anything. So.

Bridget Shapiro
Yeah, yeah. I was just going to say that the airport ones or ATMs, right?

Casey Naiduk
Sure.

Bridget Shapiro
Getting tickets, paying for parking, that sort of thing. All that. All of those are considered kiosks, right?

Casey Naiduk
Absolutely.

Garry Harstad
Yeah.

Bridget Shapiro
Great.

Okay, well, then this kind of goes along with the next question. So what should be taken into consideration when developing any of those things? And outside of the normal considerations of developing, say, a website or a software program?

Casey Naiduk
Garry, if it's okay, I'll jump in.

Garry Harstad
Yeah, go for that.

Casey Naiduk
and kind of just piggyback on some of the points that you made in response to the first question. So.

Garry Harstad
Right.

I was going to say I pretty much answered that question incorrectly with the first question. Just piggyback up of that.

Casey Naiduk
Well, hindsight's 2020, but yeah, I mean, there are some in some ways the same principles apply, right? We want things to be to be findable, usable in whatever ways people might interact with those, with those that content or functionality. But when in a mobile environment we have some special consideration. So Garry mentioned the smaller screen size... Touch is another kind of biggie, right?

And so how do we ensure that... How do we ensure that things are accessible by gestures and touch and things like that? And further than that, how do we ensure that people can do things like explore by touch without perhaps inadvertently making selections? And that becomes kind of a big kind of a big issue sometimes in touch screen kiosks, which a lot of them are.

But by and large, the same accessibility principles that apply to websites apply more broadly. We want to use colors with strong contrast. We want to make sure that things are built in a way that they have a good, you know, strong semantic structure. And we want to do our best not to ever interfere with the assistive technologies that somebody might use.

And those remain true. Those points remain true regardless of where the content lives or how it's accessed. What say you, Garry?

Garry Harstad
Yeah, no, you're exactly right. And one of the points about the viewport being such a smaller size does lead to the idea of what you search is to touch targets, which also brings up some potential differences between iOS and Android. But in general speaking, yeah, you need enough space around a touch target so you're not mashing the wrong things or pressing the button buttons or worse are not able to press the right buttons because they're too small, right? Which basically renders the actual functionality pointless, you know. So yeah, absolutely right. I think a common example of a mobile versus non-mobile is the idea that whenever you see a hamburger, hamburger menu, those typical three lines, you know, you're looking at a restricted width device and that that may or may not be that much fluidly or as a fixed viewport decision making system. Because as you said, kiosks, they tend to be fixed in their in their orientation so that they portrait or landscape.

Usually the software running on that will often just take into account, oh, that's the width we have, that's the height we have, here is the information you want to store there. So they’re one of the major things you take into consideration of any size because of touch target and size because of display material,

Bridget Shapiro
That makes sense.

So how do the differences in iOS versus Android operating systems affect developing mobile apps? Are there things that need to be considered? Or...

Garry Harstad
Yes, yes. I mean, a lot of that is technical, Bridget, right? So I mean, a lot of the issues are, you know what do I have to compile or how do I have to build a program to then get deployed within a container that then gets put on the App Store? A lot of technical differences. But as Casey said originally, there's not really a an accessibility major difference or paradigm shift between the two.

It's more like, how do I make an Android thing the right pixel size versus an iOS thing the right pixel size? And some of the considerations around that are density of pixels on the screen, right? So iOS sometimes has a, you know, retina display, so it might be three times the pixels of another one. You just got to make sure that overall things like touch targets are within your minimum sizing for whatever device you’re on. So for instance, a 48 pixel, a standard 48 pixel button might appear or render very differently on an Android device than it does iOS. So you've got to take into account the density of pixels. And again, that's where human usability testing will find that out anyway. If you think you've done it right, you'll find out what with the human.

Bridget Shapiro
Right.

So each of these operating systems has its own accessibility features, right? Or not features, but software built in.

Garry Harstad
Which you don't want to break, as Casey mentioned. But also you don't want to rely on. You know that makes sense.

Casey Naiduk
Yeah. Good point.

Garry Harstad
You don't want to break things or break AT technology.

Casey Naiduk
Garry to that, to that point, though, I do feel like another point is warranted, which is I think there is a conception sometimes and I think it's a misconception that because iOS and Android devices have accessibility features built in like voiceover on iOS, like talkback on Android, that, you know, if you've got an app in the app in the, you know, Apple App Store, for example, that it's going to be accessible.

And we know that that's not true, right? So there are two considerations here. Yeah, we don't want to interfere with assistive tech people might use. We don't want to break that. Right? We don't want to ever disable that. But that doesn't mean that we're then kind of let off the hook from making deliberate design and development decisions that will enable those user agents to access the, in this case, the software properly and convey that to their users properly.

So it goes both ways. And then, Garry, you would know better than I, but I think in iOS and in Android, those can potentially be quite different, you know, coding styles and languages. And they may or may not be the same people I think, who would be skilled in, in building out those programs. Is that is that kind of right?

Garry Harstad
Yeah, there's definitely I mean, let's let's take one example. The most common development system for apps in general is Mac based. And this is mostly because the systems to deploy

containerized apps was more conducive to Macs. And Mac is closer to a Linux system, which those containers tend to run on. So it's getting a lot better. Things like the Docker system, which is a self-contained or containment building system, you often deploy an app whether you build it in Windows or Mac to a an operating system agnostic container.

So that means that technically speaking, you can put code into that container from a Windows device or over a Mac device. The only thing that’s OS specific is how you deploy it, because certain scripts, you know, aren't available on Windows that they might be on a Mac. So but you're right though, if I'm a good Android developer, it doesn't mean I'm any good at dealing with iOS related stuff potentially.

And again, not all apps are even based on HTML which most of the spec is for. So if you're writing an app that renders its own screens and doesn't use actual HTML, it's up to you to follow the letter of WCAG and kind of the equivalent, you know, the functionality you're looking for. And it's important to notice, too, that a mobile app isn't always a mobile app.

It depends on the context. If you take a mobile app, fix its rotation and put it in a kiosk, the presumption is they're only going to operate your app unless the app exposes implicit operating system accessibility tools. So a lot of apps aren't going to let you access the operating system or those in tools anyway, or even the web because it's a security concern.

So whether it's for the phone or kiosk can be a major differentiator on how you deploy and what features are available, such as an operating system level voiceover or something like that.

Bridget Shapiro
I haven't heard this too much recently, but I feel like in the early days of my foray into the accessibility world, some of the clients got frustrated from that and were like, well, we're just going to create different websites, one that's accessible and one that's not, and we'll create different apps and all these things. And the idea is never that right?

That's still the case. You really just want to cater the one to make it accessible to everybody. But and again, I haven't heard that so much lately, but it was the thing that I was hearing in the beginning where people are like, well, we'll just create a new site that doesn't do much that's accessible to everybody.

And obviously not...

Garry Harstad
Back in Section 508 days, they actually said an alternative to the website is good and then that got phased out with WCAG. So I think it's mostly about sitting on that hope that they can do something different. But the accessible side and that's not the case. We want an equal playing field, not a different one.

Bridget Shapiro

Exactly. Exactly.

All right, good. All right. I'm going to move on unless than anything else needs to be discussed on that topic?

Casey Naiduk
I think we're good.

Bridget Shapiro
Great.

So it's kind of along the same lines. A lot of these questions seem to be. But is there an operating system that might be more accessibility friendly, quote unquote, or is that not a thing?

Casey Naiduk
Well, I think it's a I think it's a fair question. And I think this is going to sound like, well, unless Garry has a brilliant answer. I think this is going to sound like a like a softer, weak answer. But I mean, every word of it. Accessibility, needs of people are going to vary tremendously. Right? And so and so traditionally, you know, Apple and by extension, iOS has had the reputation of maybe having packing a little bit more accessibility punch, having some more features built in voiceover is a great screen reader, you know, some great contrast options, some great, you know, text resizing options, all built in. For many people with different types of disabilities that has made Apple the mobile device of choice or the tablet device of choice. But from my understanding that gap, that gap is closing every day. Android is is releasing new features as well. And it's really not one size fits all, right? So what's more accessible for one person may or may not be what's more accessible for another.

Right, Garry? I mean, that's yeah, Oh yeah, it's kind of a cop out answer, but at the same time, this isn't a monolith, right?

Garry Harstad
Yeah, Yeah. And I think it's important to distinguish that this question particularly is about the, the OS itself, not about developing apps on it. So if you were going to pick up an Android device, an iOS phone, which would Casey think is generically more accessible quote and I think Casey is right and Mac'll have the edge right now. But that gap is fast closing and it's very interpretive and so on and so forth, but ultimately has nothing or very little to do with actual app or kiosk code generation.

Bridget Shapiro
Gotcha.

So iOS by a small hair. But, uh...

Garry Harstad

At least out of the box.

Bridget Shapiro
Yeah.

All right. Got it. So what changes in the development process when developing an app over a program designed for a desktop? Like what? Or are there what happens? What are those changes and whatnot? Does that make sense? That question?

Garry Harstad
I think the question makes sense. The answer might not. You know?

Casey Naiduk
I think that's our that's our specialty, Garry.

Garry Harstad
So to me, a developer, a developer doesn't necessarily care about the difference. What he cares about is where it gets rendered, you know, what a user sees. So if your tech stack, should we call it whether using a combination of libraries and frameworks have some tools that make it easy to deploy a certain way or to a certain device than another one that generally dictates your software development lifecycle?

And as I said, because Mac tends to be a little more popular in app development, there tends to be more tools related to that. So what changes tends to be your tech stack or your tools used to build the app in the first place. The end result of the app is pretty much what we've already discussed. You know, the output is going to be dependent on the design, dependent on the viewport and dependent on how it's used and whether they have access to our space functions.

So what changes? Your entire development lifecycle. So if I'm doing something specific for Apple, it will be very different than I'm doing something very specific for an Android app versus on coding for Windows desktop. They're all about the tools you use to generate the package and then deploy the package because each of those devices have their own package deployment mechanisms.

The, you know, can be mostly about that. And again, as I said, it wasn't a great answer, but that's the nature of the answer. We can go there.

Bridget Shapiro
Got it.

Anything else to add, Casey, or should we move to the next one?

Casey Naiduk

I feel like I've got more questions than answers for that one. But yeah, you know, sometimes it is the case that mobile operating systems, mobile operating systems are updated maybe more frequently than desktop operating systems and even technologies and mobile devices are sometimes occurring updates and updates to the iPhone, for example. What is that every every year, sometimes more quickly than that and like kind of like the desktop equivalent?

And so what's going through my mind is as those operating systems update, as the actual, you know, device or form updates, I'm wondering if that if that wouldn't increase the the frequency and sometimes urgency with which you need to kind of perform tests and maybe push out fixes or patches. Right? So Windows doesn't change all that terribly often compared to, let's say, mobile operating system.

And Gary, I mean, you even know better than me, but some of those some of those mobile operating system updates may or may not break or change what's happening in the app. Right? Or am I overthinking that?

Garry Harstad
I think it's a bit of both. You're right in that a mobile device definitely updates more frequently and has a much greater chance of introducing a new feature that you or your customers may presume you're going to leverage. Outside of that, if you coded your app right, then operating system level changes shouldn't affect the app. Unless fundamentally they've removed something about the feature set. Because like I said, you really don't want to develop ideally related to the operating system's implicit accessibility.

You want to make the app accessible by design by itself, no matter what the operating system. And ideally most apps are operating system agnostic. At least that's the goal. And we're slowly getting there. But like the when the browser wars happened and now we're kind of coming to this standard where most browsers operate in the same way now. I really dislike the idea of having to build very specifically to Android versus having to build very specifically to Apple.

That's not going to change in the near future. But what we do hope is that there's less and less, as Casey mentioned, breaking changes. One, because your app hopefully won't of leverage things it shouldn't have. And two, you hope the operating system didn't rip out from underneath your core principle you designed this on. Outside of that, yeah, it shouldn't matter.

Fingers crossed. Right? I'm sure there's plenty developers that are yelling at me right now for the spring thing. I can give you a hundred reasons why it does, but I'm not aware of that many of them.

Bridget Shapiro
Well, that's why we have them all muted.

Garry. No worries.

Bridget Shapiro
All right. Awesome. So this just might be repetitive. I'm going to throw it out there if you guys want me to just move on to the next I will. But the next question is really about what processes are unique to accessible mobile app testing? So we talked more about developing, but maybe this is more focused on the testing?

Of mobile apps.

Garry Harstad
There are tools that emulate, right? That there are automated tools that emulate testing your app quote within a emulated iPhone, emulated Android. But honestly, they're never going to replace the actual value of a human also testing it because it's just something that you can't automate no matter how good you think your AI is.

Also, emulation just isn't physical. It's just a helpful copy of what the new latest iPhone is doing. But there are tools out there. So what's unique about accessible mobile app processing? Again, as I mentioned earlier, it's more about the tech stack and the tools you use. That's what makes it unique to deploying to a mobile device and the testing related around that would usually be some combination of using a cloud based tool to test your app before you deploy it, and then hopefully go through some constant integration because you know that software development lifecycle where we're testing accessibility before we deploy and then hopefully the final cherry on the cake is humans then test and

audit and verify that all of the important workflows that you just put into place are actually accessible and functional. Because sometimes functional isn't the same as purely accessible. You know? Or how difficult it is a blurry factor and not necessarily a pure straight black or white violation? So yeah, to me, again, I think the answer is tools. The tools you use to develop will involve some kind of unit tests or cloud based emulation of your app.

But the final test should be a human auditing.

Casey Naiduk
Absolutely. If I...

Bridget Shapiro
I love

Casey Naiduk
Go ahead, Bridget.

Bridget Shapiro
by human testing.

Garry Harstad

[inaudible]

Bridget Shapiro
Manual testing, I think. Yeah.

Casey Naiduk
Yeah.

Casey Naiduk
And can I, can I add a little bit there.

Bridget Shapiro
Yes, please do.

Casey Naiduk
Yeah.

Yeah. There's not going to be, there should never be, right? We shouldn't be conceiving of of a testing strategy for anything including mobile apps that doesn't include the human or the alien manual review. At the end of the day, it's people that are using these things and so a person, ideally many people, but at least a person trained in this area needs to test it.

And so when you're testing a mobile app, we've touched on some of them, but there are some additional considerations that you have to keep in mind. And so one is that, you know, we've touched a little bit on iOS versus Android, but ideally you're going to test on both, right? You're going to test with both screen readers, for example, Talk Back and voiceover.

The reasons are you don't want to do just one or the other is because they will behave a little bit differently. They are different programs on different operating softwares, and sometimes... This may be common knowledge to some, this may be new to some, but sometimes different screen readers, different assistive techs, will do a better job or a different job at confidently guessing.

For example. So let's say there's a button that has a bad label or a missing label or something. Sometimes Talk Back will be bold and say, Here's what I think that thing is called and voiceover won't touch it, right? Sometimes they perform differently, so you'll want to test with the gestures, the expected behaviors, the kind of common commands on both would be one point.

Another point that I would add, and this often gets this often gets overlooked in mobile testing, and I understand why, but I would strongly encourage anybody listening, anybody watching to reconsider introducing keyboard testing into anytime you have you’re testing a mobile website or app. And by keyboard I mean I mean a keyboard. A Bluetooth physical keyboard. And a lot of times we think of kind of swipe gestures or touch controls as the key, as the as the mobile equivalent of maybe like a keyboard tap stops or keyboard use.

And a lot of times that's true. But if you have, let's say, a mobility disability that prevents you from being able to or wanting to use, let's say, a mouse on a on a on a laptop or a touchscreen on a laptop, that's not all of a sudden going to magically change because you're using a smartphone, right? So test with that actual keyboard.

And if anybody's scratching their head at this, yes, your apps and websites can and should work with physical keyboard. So it's often overlooked. I understand the reason why, but I would encourage, I would encourage it not to be

Bridget Shapiro
Right.

Casey Naiduk
Yeah.

Bridget Shapiro
Great. Very true.

All right, good. So let's see. We'll move on. What specific pieces of assistive technology should be considered in the development and testing of mobile apps? Again, all of these are kind of intertwined, these questions, but...

Garry Harstad
Yeah. I'll echo what Casey said. Many people will be surprised to know that their apps, mobile apps, are accessed by a keyboard and even might, you know, as Casey said, as long as somebody has a dexterity issue, that's not going to change because they mumble on an iPhone. The reality is they kind of plug something into it, whether it be a trackpad, a mouse, a keyboard or some form of input device, which they move with their head or whatever it is, you have to test keyboard on your mobile apps.

And if you're not doing it, you're already failing the deployment process for accessibility. So if you don't think you can do that, which can be tricky. Again, if you write a magical app that doesn't use HTML for instance, it’s like how the hell do I test this? What you need to do is find the human. Find that human again and say, hey, they will know whether it is working because they're going to use a screen reader is going to announce things, but mostly or not at all, they're going to be at a keyboard, navigate to it or not. It'll be a confusing hierarchy or not, and a lot of that stuff doesn't manifest when you visualize it and you just look at it. So physical testing and plugging in keyboards and use of mouse even for mobiles are definitely something you should do as Casey said. You know, things like focus visibility often gets overlooked or even tablet because they think, oh, just press it.

Yeah, but there is a guy trying to tab as well that you're not looking at you, know? So again that goes back to our principle that all development about accessibility should be the same. You know you should test for the same things. But it is important to note that when you make a mobile app, it's seemingly instantly forgot that most that mice and trackpads and keyboards are actually going to be used on that.

Bridget Shapiro
Right.

Garry Harstad
So make a point of testing it.

Bridget Shapiro
Great. All right. So definitely focusing on the keyboard and all those other tools is a good idea, it sounds like.

All right. So I think we've talked about this one, so I'm going to move on. This one is about the deployment process between apps and desktop programs. I think we've pretty much covered that or unless you guys have anything else to add, how do they differ? I think we've already...

Garry Harstad
Yeah.

I mean, the different editing kind of going on, but they can be, you know, there are frameworks that are good for one and not as good for another. You know, a common framework is Node.js, for instance, and then using a various set of libraries to create a user interface. So a lot of it is really again, how you deploy it and how you build the application.

So we have kind of gone over that. So it is based upon deployment target should we say, you know?

Bridget Shapiro
Node.js, meaning JavaScript right?

Garry Harstad
No, there's a process that there's something called Node.js and it's just basically, yeah, it's like a JavaScript framework that gives you access to back end functionality as if you had a server. And it's becoming a very popular modern way of deploying apps. But again, you could use any library system to do that, but usually you do it to a Docker container which would be OS agnostic.

So in theory, the difference between desktop and apps could be that a desktop app focuses specifically on being a Windows app versus being a agnostic Docker based app that could run anything. But other than that is not a huge difference really just what tools you work with. I probably sounded a bit rambly, but that's it.

Bridget Shapiro

That's all good.

Casey, you feel good with that? That I think we've covered that well? Or is there anything else to add?

Casey Naiduk
Yeah, I mean, I think I don't have a whole lot to add, but I guess I guess one consideration is that sometimes when we think of desktop apps as opposed to other web based app or mobile apps, if they are like Garry brought up like a Windows application, if they are kind of like a standalone Windows app or something.

Sometimes when you when you decide you're going to ship something, you better be pretty darn sure it's good to go because getting users to accept or turn on any updates can be can be a challenge. Right? And there's some logistics and potentially cost involved in that. Where it could be potentially easier to push out updates on like a mobile app or something because people might get that nice little icon that tells them there's an update or something like that.

But, no, I think I'm not going to beat that horse anymore.

Bridget Shapiro
Gotcha.

All right, good. So in in apps with apps, what's the best practice for recording user feedback? How is it best to respond? In the app? In an email? Other? What's your thoughts around that user feedback?

Garry Harstad
I mean, my opinion is in general, you would you would want to keep people at the digital asset their own. So I'm an advocate for saying if you want to offer feedback from an app, give them a thing in the app. Don't point them to another website, make them go somewhere else, find out the different ways of accessibility, controlling that, you know, make what you want to happen within the app.

Having said that, there are reasons why people won't do that or might not like doing that, but that's my general principle on it. What do you think, Casey?

Casey Naiduk
I agree.

And there are a lot of there are a lot of product specific business specific marketing and metric specific reasons why you want to keep people on that thing. I'm in agreement with that. What I would add is to give people options. So whichever mechanism you deem is your kind of primary or best way of gathering user feedback, give people options. Because what they're comfortable using and what they're able to use or able to use confidently is going to vary.

So don't offer, don't offer just one option. Don't offer just...

You know.

Don't offer just a text based or a voice based or a phone call based or anything based option. Give people give people options so that so that they're able to use it.

Garry Harstad
That's a good point. Casey brought up a really good point where, you know, I could have all the best intentions of the world and create a nice contact us form, which happens to be ironically inaccessible. Right? So there should always be a if you can't use this email us, because whether you know it or not, most people with, you know, impairments have figured out ways of navigating the common apps we all use, such as email and word type documents.

You know, they already have their own ways to get around it, so don't restrict them to having to do it this one way. You just going to create a huge obstacle and frustration block for them if they can't use it. So to Casey’s point, offer more than one way out

Bridget Shapiro
Options. Yep. Okay.

That makes sense.

Casey Naiduk
Yeah. I would add... Bridget, I would add one thing too, which is I think this is I think this is sometimes a big miss, which is after feedback has been provided. So whether I click the button and said, Yes, I love your service, no, I hate your service or have submitted a form or I've done something, what happens next?

Make sure that that's available and accessible to people in different ways. So if there's a status message or something, make sure that assistive tech users will look at that status update. If there's a success chime or bell or something, make sure there's a visual indication of the same, right? Don't neglect the post feedback, you know, accessibility because it's...

Bridget Shapiro
Right. Yeah.

Casey Naiduk
Yeah.

Garry Harstad

I guess the running takeaway there is if you're giving them an out to get feedback, if nothing else, make sure that process is 100% accessible. Because otherwise you've just left them in this box of can't even report the problems, you know, and you're kind of stuck.

Bridget Shapiro
Right. Agreed.

All right, good. So mobile operating systems update more frequently than desktops. That is the beginning of this question. But Casey, you just said the same thing a little while ago. So we all agree, you know, we are always getting those updates to our mobile phones, it seems more often than desktops. How should testing accommodate for that? You did touch on this a little bit, Casey, but is there anything else that specific to the specific question that needs to be addressed?

Casey Naiduk
I would say not necessarily specific to the question, other than it's important to establish kind of a testing and auditing cadence and to remember that accessibility is not a one and done and the frequency with which mobile systems update should hopefully drive that point home. So if you've built an app today and you've tested it, you know, in the summer of 2023, you've got to, you know, whether it's every three months, six months, one year, whatever it is, you've got to have a system in place and a tracking mechanism in place to make sure that you're keeping up with that, because technology changes, that means necessarily the accessibility of your app over time will change.

The way people access. It will change. And so nothing specific to the question other than you know, it's not a one and done.

Bridget Shapiro
Yeah yeah, I agree with you. I think you know I've worked with a lot of clients and those that are more successful than others are those that go through the audit and then continue to have retests done in that type of cadence that you just mentioned. They're checking to make sure that what they've the remediation that they've done after they learned what was wrong during the audit was done correctly, deployed correctly, and now is it accessible.

And then once they get through that, then of course the maintenance. You want to keep checking and just double checking. So those are always the more successful clients. We did have some clients that strictly wanted an audit and that was it. Tell me what's wrong with it and we'll fix it and one and done. And obviously those are the ones I saw having a little bit more trouble with keeping up with their accessibility of course.

So yeah, I think that cadence is definitely not just important, but seems to be integral to the success of the clients I've worked with.

Garry Harstad

And I think that goes back to the point of trying not to use implicit operating system features because they're the ones that are likely to get changed or break. If your app, for instance, is using straight HTML than an operating system update shouldn't affect that. So again, you're the only one that knows what features you're using and leveraging. If you're leveraging an operating based system, you should subscribe to that operating level bulletin notes about future releases, and that's you as a developers job to make sure you on top of those magical changes.

Bridget Shapiro
Gotcha.

Casey Naiduk
And that's one of the that's one of the kind of our core principles, isn’t it Garry? Robust. So we would we would hope and ask people to build to spec standards but maybe not necessarily to specific technologies or user agents at any given point in time. Right? Because that point in time is going to pass.

And we need something that's going to be almost agnostic, as the good word that Garry used, and be able to adapt to user agents that we might not even know exist yet. But if we've built something to be accessible, they should be able to do their jobs right?

So.

Bridget Shapiro
Right. And you just alluded to robust being one of the principles. So that's that POUR, right, that people talk about. P-O-U-R. You are perceivable, operable, understandable and robust. I wanted to pull it up so I didn't mispronounce any of those words, but that's one of the four that all these different checkpoints under the WCAG guidelines all fall into.

Right?

Garry Harstad
Right. I think you brought up a good point. That's a good rule of thumb I use. The P-O-U-R. All of the one dot criteria are P. All of the two dot criteria are O. So you know that anything regarding robust is going to be a four dot specification in WCAG. So people often don't notice an association to the POUR and the extra numbers.

But it's worth if people remember POUR, you kind of can remember the nature of the WCAG success criterion.

Bridget Shapiro
Great. Good trick.

I did not know that. So, thank you. I just learned that, too. It all... I thought they all kind of had to do. Yeah, I'm sure that there was an...

Garry Harstad
There’s an actual logic.

Bridget Shapiro
Yeah. No, that's great. That's good to know. So how can you measure success metrics for an app's accessibility? We've talked about testing. Of course, that's important. But how do we measure this? Like is there going to be this magical I'm accessible now or is it always that process? And how do you measure that?

Casey Naiduk
So.

No, there's not going to be a magic. I'm accessible now and there are I mean, the metrics that you care to capture, you know, your mileage is going to vary if you are a small company with, you know, a relatively low number of assets, the metrics that you care to capture may be, you know, for this one specific app, how are we doing over time?

What's our number of accessibility bugs? What's our volume of maybe negative user feedback or questions or struggles using the app? And you might, you know, might benchmark and compare it against itself over time. If you have a large portfolio, you might be comparing websites or apps against one another and how they're trending over time. But typically, I mean the ultimate, however you want to bucket those metrics, in my opinion, choose what makes sense to you, but your goal should kind of remain the same, which is how do we make this thing wonderful and enjoyable and intuitive to use and whatever feedback mechanisms you think are going to get you there and get you there most reliably, put your eggs in those baskets. In that basket.

Bridget Shapiro
Yeah.

That makes sense. So improvements, it's really about kind of defining your own.

Your own. I think it's defining your own. There is no standard on measuring success. I think measuring success is improving it as time goes on if there are issues. Right? So how do you define that? Of course, it's going to depend on the team and the power you can have put behind that. But I have found that our clients, that they had a clear goal, they had those clear, okay, this this quarter, this run or whatever we're going to call it, the sprint, if that's what they were doing, we're going to focus on the user flows.

Those are so important. We need people to be able to successfully purchase from us. So again, we tested those types of user flows, use cases, not just static pages. It was about going from one thing to the next, and that's all important as user behavior. So, you know, some of our clients would say we're going to work on those user flows first, and that was our first measure of success.

Then we're going to fall into those things that might be a harder lift down the line. But you start seeing the number of issues kind of dwindle and dwindle, and that was their gauge for success. As you do retests and remediation phases.

Casey Naiduk
Yeah.

That makes sense, Bridget. And that potentially introduces another kind of type of metric that companies may be interested in using, which is, you know, over time, as the number of potential issues decreases, you may be able to and may want to measure kind of the manpower in supporting or, you know, achieving or sustaining accessibility, hopefully over time that that's on a downward slope, too.

Right?

Bridget Shapiro
Right, right.

Garry Harstad
Yeah. It's also tricky because metrics are just metrics. Just as Casey said, they just made up points that you care about. And there really isn't a magical way to automate accessibility metrics. A success criterion, as you both alluded to, is, is my workflow functioning? Am I still getting people going through it and have my numbers dramatically dropped or have they dramatically increased?

That still doesn't tell you if you're isolating disability product concerns though, you know? So again, if you actually were about accessibility metrics instead of just app metrics, in general, accessibility metrics can only be done with periodic audits because you need a human who has that issue or using that assistive technology to say, yes, you're still completely functional. Until then, you just looking at an aggregated numbers game, which is valuable, it's just not necessarily accessibility centric.

Bridget Shapiro
Right.

Gotcha. Good.

So I am looking at the time I think we might be able to take on one question and final as our final question, but feel free to after we kind of go through this last question I'm going to present, you know, if there's any kind of closing remarks just about accessibility of mobile apps or kiosks you want to throw out there, please do.

So the last question I think we might be able to get to today is are there other things that should be considered when developing a kiosk? And I think they more me mean, you know, you had mentioned earlier that it's a it's a physical thing. Like what are the things do I need? We all think about digital accessibility, of course. That's what the world we're in. But there's more to that with a kiosk. And what might those be?

If you can touch on that?

Casey Naiduk
Go ahead, Garry.

Garry Harstad
Okay. I was going to say that, you know, it kind of ties back to the mobile versus not mobile in that you know, if you imagine what are the ups and downsizes of using a mobile device and actually writing for it. What is this device have? It has it has a battery. It has a small viewport. It has memory.

I might get a call on while I'm using an app, you know? So these cumbersome principles that were written down here work were for a mobile app, we generally do functional testing, performance testing, memory testing, interruption testing, installation testing and usability testing. So usability testing tends to be the accessibility, close to the accessibility side. But with a kiosk, for instance, you can imagine that's plugged in so you don't have to do the battery full power level testing.

So depending on your platform and your final orientation, it'll determine what technical things you're going to be testing for periodically. You know, like I said, you really probably shouldn't have to worry about interruption of a kiosk because it shouldn't be accepting phone calls while you're using it. Right? It shouldn't have a power concern because it should be being plugged in the whole time.

But outside of those, you know, that's that really speaks to the difference between mobile and non mobile. The only difference being with a kiosk, you can eliminate a few of those because it's plugged in and doesn't accept calls. So that's all I wanted to say about those areas.

Bridget Shapiro
Gotcha.

Casey, do you have anything you wanted to add?

Casey Naiduk

Yeah. And there I'm glad to get that question because there are a number of different ways to take that response, I think. But yeah, kiosks, kiosks exist in the physical world and so that kind of necessarily introduces some physical accessibility requirements, some of which are, you know, ADA driven, some of which are kind of common sense and good practices.

But for example, where is your kiosk positioned and mounted? That's something you don't have to consider with the mobile app, right? A mobile app exists here. A kiosk could exist in a crowded space. It could exist on an upper level, on a lower level, it could exist in a noisy space. So all of a sudden you've got these additional physical environment considerations.

So what is the path to access it? Is the path to it accessible to people, including perhaps wheelchair users? Around the kiosk itself there are actually some specific requirements that the depends exactly where it is, but around the physical space itself. So you're required, I think from the ADA 2010 update, I might have the measurement off, but I think it's a 30 by 48 inch clear space in front of the kiosk.

I think that's a requirement and I think that can go in either direction. Yeah, the idea there is so a wheelchair user or somebody with a companion or something else can have the physical space in a room to actually access that thing. Another thing I mentioned I think is the height. So there are some specific requirements around how high or low kiosks are able to be.

I'm fairly sure ... I'm going tp have to double check. We might have to send out an email or something if I got these numbers wrong so people don't go ahead and build the wrong height kiosks. But I'm pretty sure, for example, the maximum height for things that you have to be able to touch I think is 48 inches off the ground and I think the minimum is 15 inches. The reason there is so whether you're standing, whether you're seated, you have reasonable access to it. So physical considerations are important, environmental considerations are important. And then additionally, do I have time for a couple more points?

Bridget Shapiro
Yeah. Okay. Please. Good stuff.

Casey Naiduk
You know, additionally, you know, Garry, Garry mentioned correctly a couple of times that one of the primary differences between using a program in a kiosk setting is that you don't necessarily have access to all the features and functionality of the operating system itself. Right? You're closed off in that program only. What that means then is we might have to we might have to find ways to still allow people to get the assistive tech and features that they need access to.

Bridget Shapiro
Right.

Casey Naiduk

So it's still a requirement, for example, for a touch screen kiosk to have, you know, an alternate output and input mechanisms. Right? And so sometimes that is cared for by traditional screen reading technology, which we might, you know, which we might know is something that kind of lets us navigate and read and make selections. But sometimes there might be other ways of achieving the same.

And it doesn't always necessarily matter how you care for that, but you need to care for it. If we think of other certain other there certain kind of physical or kiosk specific features. A headphone jack, right? That's expected on ATMs. That's expected at airports. But what happens when you plug your headphones in? There better be there better be something on the other side that's ready to communicate to you.

Right? Sometimes we'll see kiosks that have the potential or the capability of offering, you know, kind of accessibility functionality. But those things have been disabled or haven't been proactively enabled. That's a miss and that's a risk. Lastly, I guess keyboard. So keyboard seem to be my theme for the day. But sometimes kiosks are very like, you know, one or two click or one or two interaction specific.

And for those types of things, you might not need, let's say like a full keyboard. Some things require a good amount of user input, right? So if you're at the airport, if you're somewhere where you have to enter your full name, your full address or something, you can't just be a touch screen that's available for people to do that.

Right? Like one of the things I mentioned at the onset of this call is there's a requirement around kiosks that you're able to explore touch screen without accidentally selecting things on a touch screen. Well, that's super hard to do because when you touch a touch screen, things happen. Solution to that are other tactile or alternate input devices like physical keyboards.

So I think the takeaway is depending on the industry that you're in, look and see if there are specific rules and requirements, right? If you're an airline, for example, there are specific rules and requirements. If you're you know, if you're a public accommodation that is that falls under the umbrella of the ADA, there may be specific expectations and requirements around the physical spaces that you need to care for.

Ask those questions, find those answers, Don't find out the hard way. And yeah, that's it for me for now.

Bridget Shapiro
Well, a real life thing I can equate to this. It's not digital accessibility, but post office. When I drive up to the post office to put mail in, I drive a sedan. I feel like all of the mailboxes now are for people in SUVs. So I'm constantly unbuckling. Having to open my door and put go up.

It's the same idea. You know, you have to be accessible to everybody. Not everybody's in an SUV and driving. So, you know, it's the same type of idea, Right? It might not be fully under the same topic, but it's my real life experience just yesterday and it's super frustrating. So, you know, I'm hoping that what we've done today is just our part in helping everybody kind of be cognizant of making this more accessible to everybody, everything more accessible to everybody, making it an equal playing field.

To your point, Garry, I believe you said.

Garry Harstad
I see it, but Casey brought up another point, which I think is important to bring up now, is that if you do use a keyboard particularly, you're going to probably have a virtual keyboard in use or a touch screen scenario. You can actually, per operating system, choose to have the different kinds of virtual keyboards to come up.

One thing you don't want to do is if asking for a phone number, given the phone code, a keyboard, that's just annoying and confusing. So if you are going to use virtual keyboards, there are ways to invoke the correct type, you know, and some even have nice symbols and easy ways to complete your email address and things like that.

So I would say additionally, if you're going to have virtual keyboards, make sure you try and leverage the keyboard type for the type of input you're trying to use. You know, phone numbers should only show you numbers mostly right? The only other thing to consider is that developers of kiosks tend to not like people to plug things in, but you can imagine what kind of security issues the USB being plugged in can have.

So although it is an idea, I don't think you'll see it often in practical applications yet, although it should be, there doesn't seem to be a security way to kind of solve those hackers. Yeah, yeah, yeah. Great.

CONCLUSION

Bridget Shapiro
All right. Well, then, if no other points want to be brought out, have we kind of exhausted everything we wanted to talk about? And we've, of course, exhausted time. So that's our ending point here. But I really want to thank both of you, Casey and Garry, so much for everything today. I appreciate your time, your thoughtfulness, and, of course, your expertise.

And to everyone, I'd like to just remind you that the event is being was recorded, is being recorded and will be available for viewing later tonight along with the transcript. And again, you're going to receive an email from Lori with that recording and transcript as well. I'd also like to once again thank our sponsors, Bureau of Internet Accessibility and of course, Clusiv, for supporting today's event.

And I'd like to invite you all to join us for our next event. That's going to be held on Tuesday, October 24th, 1 PM Eastern. That's going to be around creating a sustainable digital accessibility plan. So that's sure to be a great one as well. So thanks everybody. See you them and have a great day.

Casey Naiduk
Thanks for having us.

Garry Harstad
Bye bye.