

### 3Play Webinars | 05-11-2017-UX

---

PATRICK LOFTUS: So thank you all for joining this webinar today, entitled Usability and Accessibility Have a Conversation-- How Accessibility and UI/UX Teams Can Collaborate for More Inclusive Products.

I'm Patrick Loftus from 3Play Media, and I'll be moderating today. Today I'm excited to be joined by Jiatyan Chen, Online Accessibility Program Manager at Stanford University. And with that, I'll go ahead and hand it off to Jiatyen.

**JIATYAN CHEN:** So good morning. Good afternoon for those on the East Coast. I'm Jiatyan Chen, and I'm really happy to be able to share this-- share some of my insights with you on both usability and accessibility. So we're going to talk a little bit about who I am and why I'm doing this presentation, and then we'll get into the meat, which is the how.

OK, so who am I? For the most part of my working life, I've been in higher ed. And I'm currently the Stanford Online Accessibility Program Manager. And that's pretty much a job that consults on accessibility policies and services. And we look at projects, and I do various communications.

But before that, I was at Michigan State University, and I was in a role with multiple hats. One of them is web accessibility coordinator. And I was also the IT manager and project manager. And prior to that, I was a user experience designer and instruction designer. And I worked mainly on building online courses and interactive learning modules.

So that went way back. And it was when-- it was around the time when I started building learning modules that I ran across a book called *The Psychology of Everyday Things* by Donald Norman. And that book has been reprinted multiple times, and is now renamed *The Design of Everyday Things*. Now, what that book taught me was that the problem is not on the user. The user shouldn't feel guilty about failing to use a product. It's actually the designer's job to communicate the importances and provide the necessary feedback, and prevent the users from making mistakes.

So for example, the biggest dial, the frontmost dial on a stereo, should really control its primary functions of making the stereo sound, particularly when the user is distracted by other activity-- say, driving. So I really get mad when I accidentally-- I almost always turn the dial to flip the-- wanting to flip the volume, whereas I flip the radio channel, and it makes the same amount of noise when I was driving and distracted. So usability, or the lack thereof, has mainly kept my interest since reading that book years and years ago.

So how do we approach usability? Another story is when I was moving across the country and I had to pack a whole room full of stuff into as little as possible, and pretty much turned the whole room and compress and throw away everything else, and fit it into a banker's box. And during my recycling fix, a bat actually joined me one night, late at night in my messed-up, messy room.

So I threw away a lot of the stuff, recycled most of them, and I kept-- one of the few things I kept were two buttons. One of them is a usability button called, "Know thy users, for they are not you." And the other one is, "if the user can't find it, the function is not there." And these two buttons have stayed with me. And they're now on the board next to me. And they constantly remind me of user-centered design.

And user in the sense of-- users not as mechanical monkeys pushing buttons perfectly every time, but rather, users come with their own experience. They are in their own state of mind. They are put into different situations, and they have different abilities.

It actually took Whitney Quesenbery, who's an expert in user experience, to point out to me that accessibility is pretty much an expansion of usability to include more abilities and more situations, and is fundamentally still about the users, which is why when we talk about accessibility or usability, we keep seeing the same topics coming up over and over again. And we're talking about the same thing. It's just structured differently. And we use different language and different conceptual maps.

And so for usability, during design, user designers use methods and heuristics, while for accessibility experts, they looked at observable success criteria in the WCAG. And for user designers, they design elements as a whole, design each individual element, while accessibility evaluates the experience by separate behaviors. But they have the same things, just looking through two different lenses. So the purpose of this talk is to suggest a way for these two lenses to work closer together.

I've got a lot of tips from different conferences and conversations. And the way I put it all together is to use a model by Jesse James Garrett. And that's how I organize it. I hope that it will also help you out as a map to see how to effect changes.

So the very brief overview of the model is it has five levels. It goes from abstract to concrete. And we start with the abstract at the base, which is strategy, and it goes up. It talks about scope and structure, the skeleton. And finally the surface, which the surface is actually the

layer where we get the tangible product.

But you also see that accessibility comes in really strong at the concrete and tangible level. And it kind of fizzles out around structure. And when we design a product, accessibility almost never touches the core, which is the strategy level. And that's why decisions about accessibility are usually treated as fixes, rather than changing the fundamental of the product.

So let's take a look and see how we might be able to inject accessibility further into the design process. We're going to start with the lowest level, which is the strategy. The strategy is about purpose and goals. It probably comes up with a strategic mission or vision document. It can also be learning objectives for those who are working on course content.

And we need to get the needs both from outside of the organization and inside the organization and then balance our objectives for both sides. So for example, a user might need to fill in a form with as little effort as possible, whereas the business might need to get sufficient information to sell their products. And UX designers need to balance the needs for both. And the way to do it is to hold stakeholders meetings, by drawing representatives from wherever the product affects. It could also be doing market research to find out if there's a need, if there competition, or if there are other users.

The outcome of this exercise usually is a persona, which is a close approximation of a user, but with attached sufficient background stories and characteristics about the needs for that person, that imaginary person, and the challenges that the person is facing. It's much clearer for the designer if they can design for a person than a "target audience," in quotes, which is kind of fuzzy. It's the average of everybody's personality and has no face and no story to relate to.

So another outcome of these meetings is likely to be the product's success metrics. And I'm going to show you in a bit what these two examples might look like. So for persona, this was an example from a food safety master of science program. Our persona, we gave her a name. She's Jill. She's aged 35. And she's divorced and has two young children. She's a middle manager who's looking for an opportunity to get a promotion.

And Jill, because she has two young children and she's divorced, she actually needs some flexible scheduling with her degree program. And that's a back story that we're telling about Jill.

My next example is success metrics. You can probably Google tons and tons of success metrics, probably. I found a page that has 40 or so listed easily. And they're mostly commercial. And it could include unique users that you want on a site, or if you want to make more-- if you want to achieve more sales, or time on task.

For higher ed, you could include student success metrics like student graduation rates or admission supports. Those are examples of success metrics that you can use for your project.

So how do we include accessibility into this layer? For your needs, ask more questions. Are there any other business cases for accessibility? Is there a policy that guides your university or your firm, or for your client that you're working for? Are there any financial or legal cases for accessibility, or any kind of social commitments for your organization that you want to achieve?

For stakeholders, probe more for the types of stakeholders you have. So one day, my boss and I were called to a meeting at a university's president's office by the chief of staff. And the chief of staff and secretary of the board told us that they want us to build a digital kiosk that's going to be outdoors, available to the public, and a user can browse through about 10 pages of text and media. And yes, accessibility did come up, and it has to be accessible.

It was a complex project that has a lot of-- that involves infrastructure and hardware and OS. And we have to build in a UI to sit in a totally weatherproof display, which has climate control. So there's absolutely no external I/O except for waving your hands in front of the monitor. It was super locked down for security reasons, and we couldn't add any keyboard or mouse to it.

So we went to the disability services office with this limitation. And they said, well, given all these limitations, it's kind of a pain to force the blind user to learn this new interface. Why don't you make an alternative website with the same information, and just give the website's name at the very front end of the-- just read out of the website's name at the front of the kiosk.

So we designed and built that with that in mind. And come demo day, the president herself showed up. And she liked it. And she said, can you make sure that this works for blind users, and it will speak the controls and the titles? And also make that 100 pages rather than 10 pages. So there was a total change in specs, and we had to run back and rebuild the whole thing.

And my mistake was that I assumed that the chief of staff gave me all the information that was

in the president's head. And obviously, I should have asked more questions. That was my look out for the invisible stakeholders.

And so next we go on to what you can do with the persona. You can probably pick up quite a lot of disability characteristics to add to the persona.

Remember that a persona is an average target user with a story. And designers are encouraged to use only one or two primary personas in order to stay focused with their product. So you have to kind of average in the disability stories as well. Don't make accessibility appear like a fringe case.

As for success metrics, you can probably add in, especially for higher ed, the percentage of population that you want to reach, or equivalent multiple path for the users. Or you want to improve comprehension for your courses. So those are some examples of how you can add accessibility into the strategy layer.

Next we move on to scope. The scope layer is what you want your product to do, and what you want your product to inform. It's pretty much the features and the content.

In order to get some answers to this, features and content, one activity is to do a task analysis. Use interviews, watch users to find out what they actually need, what kind of features they would like. During these interviews, you can probably come up with scenarios of how the users might be using your product, and apply these scenarios into the persona.

So for example, if you're are building a website for students, the students might want to report a computer failure with your IT website. That's the features.

And the other is part of scope is this site-- what kind of content you want to put in your site. And you run content inventory and content analysis on what you want to write. And everything kind of gels up into something called a project charter. And the project charter writes out why you want to do the project, how we plan to do it, the people involved in doing it. It also reminds us of the assumptions that we're making in the project, and what's our scope in the project.

And the project charter might prompt us to revisit our strategies of whether or not we want to do the project, and what other resource we might need for the project. And during this phase, you probably have negotiation regarding the trade-offs and requirements, and the feasibility of meeting these requirements.

So some examples of what the scenario looks like. So I'm going to use Jill as a scenario applied through persona. Jill, for accessibility, we can come up with a scenario where Jill only has two or three minutes to read a page before she gets interrupted. And so we might have to structure the writing accordingly to put the most important information first.

And for content, remember I said make a list. Make a list of all the content. That's the content inventory. And you figure out what you really want to tell your users. And you might have to combine your content and make it more succinct for the users.

So for accessibility, how do we integrate accessibility into scope? Get all the critical functions. Make a list of all the critical-- identify all the list of critical functions, and the critical content.

Figure out whether or not there is a happy path that you want to make accessible. Essentially, identifying what's critical and what's not helps you prioritize what you want to make accessible. It also helps you build the minimal viable product with accessibility in mind.

For Jill's situation again, for a scenario, you might come up with something like Jill has to be working on the course while she has her noisy children playing a game. And so in that sense, it will hint at making her course videos captioned. And she might also be-- she might have a situation where she has to have an hour commute every day. And audio versions of the course would help her out tremendously.

And as you role-play scenarios with your persona, you can identify a lot of problems earlier on in the stage. Like for example, it could be platform constraints, like if the student is supposed to report the computer outage on a website, obviously there's no computer for them to use. What other platforms may they be using to report a computer outage?

In your charter, to add accessibility, pay attention to all the assumptions that we're making about the limitation on time and capabilities. Look at the out-of-scope list and see if there's any trade-offs that you need to make with respect to accessibility.

You might even need to go back to your strategic document, back in the strategic level, to see if it's a valid trade-off of accessibility. I'm not saying that everything needs to be accessible. Just go back to the initial point of your project to see if it's necessary. And of course, go back to your stakeholders.

So for the third layer, which is the structure, the structure informs us about how the product works. It involves two big elements. One of them is how the product interacts with you. And the

other one is how the content talks to you and how the information relates to each other. And you might eventually end up also with a style guide, if there is a coordinated writers team for the content.

So an example of an interaction sequence would be your typical flowchart. You have your Start button-- a start state and an end state. And there are different decision points and processes in the sequence.

Or you might have a really complicated and bespoke interface that nobody has seen before. This is an example from a microbiology procedure called rapid strep test. When you go to your doctor with a sore throat and they take a swab, this is the test that they go through to determine whether or not you have strep throat.

And every single, probably, item on the list, and every single step is unfamiliar to you. So in the design, in the interaction design, all these behaviors and reactions has been detailed out, including what to do with the swab and how many number of drips you have to make in your well.

For information relationship, there are many ways to organize our information, according to Manuel Lima. And one of the ways to get at it is something called a card sort, which is pretty much lay out all your content that you want, and have your users or your stakeholders put them into different buckets. And it will sort out the different structures and taxonomy and figure out what's popular or not. And eventually it gets translated into your website-- different categories in your website and menus.

Another way to look at information is to, if you already have a website, go through some of the site analytics and figure out, what's the popular part, and where are people are dropping off. And try to identify those key issues and ideas. Eventually, you'll probably end up with something like a boxy, wireframe outline of what you want on your site or on your widget. And it's going to lay out the hierarchy of the sites and the data flow. And this is pretty much an information architecture document.

So how do we work accessibility into the structure? For interaction, use accessible pattern libraries. Have you ever wondered how a menu should behave when you're interacting with enter or space bar? Your WAI-ARIA Authoring Practices already has a thorough answer for you. Don't reinvent the wheel. Only spend time on the nonstandard elements, like what you do

with the test tube and the throat swab.

For content, be very intentional about your hierarchy and your sequencing, and even some kind of-- even your keywords and taxonomy. For higher ed particularly, you might want to always have a place for help and contact. This is for the grievance process that we know for accessibility. Some of the settlements are dictating that you need a grievance process.

And as I said earlier, in the writing style guide, make sure that you have-- there's consistency in your terms you use, and even get to the format of how you format a title and your link text. And the reason I say that is because of this example that I came across.

I'm looking at a learning management system. You can tell that it has some banner at the top. It has a course title. It has some tabs for the menus on the top. And then it has a modules list on the left, and the content area on the right. And the content area has some buttons for paging. And then there's a footer at the bottom.

And the instruction designers were writing documents for faculty and students about all these different areas, orientation of all the screen areas. And that's where we came across that the fact that the screen readers, when reading these areas, read them as very different things. For the banner item, the screen reader says it's a global navigation. And the module list on the left, the screen reader thinks it's a course navigation.

Now, that's not too bad. Then when we get into the content of the course, rather than different buttons on the top, the screen reader reads them as list items. And the screen reader calls the footer the navigation. So it's totally off. And it was probably because they didn't-- the screen readers was pulling from the underlying code, which had a different taxonomy.

Next layer up is the skeleton. A skeleton talks about what form the functionality will take. It's presenting for understanding and for interaction. And it involves drawing up different wireframes, navigations, and buttons and behaviors, and also involves prototyping and testing your prototypes on new users. And you can even hand it over to get some expert reviews.

So for example, wireframes kind of looks like this. It could be hand-drawn on a board. And you can talk through what each element does. It could also be very, very detailed, and laid out what each of the elements and each page does, and how they link from one to the other.

For user testing, you don't need a complete product to do the test. You can do a paper test. And here's an example of what we did.

From the designer's spec, we had a mock-up. And I printed out even in black and white. You could test for contrast this way. You print out and just grab a pencil and ask the user, what's this page about? Can you point out the places where you think this you get information about the page?

And through this exercise, we actually discovered that because there was so much gap between two headings-- there was a big heading and a smaller heading at probably 10 lines down that's indented-- that the users never relate these two headings together, when one is actually a subheading of the other. So a paper test is easy to do. And we can go back to the design. It wasn't even in code at all.

If you're going to experts for reviews, there are various usability heuristics. One of the more famous ones is called 10 Usability Heuristics for User Interface Design by Jakob Nielsen. You see that the date there is 1995. And it's still valid today. Users don't change. There was an example of a usability heuristic that we can use.

Now, to add accessibility into our skeleton layer, obviously, as you already know, the accessibility is probably loudest here. Lots of stuff to do here. You can have annotated wireframes. With the prototypes, you can already test with assistive technology. And there are multiple examples of how people laid out how we handle keyboard mappings, and how we handle errors, particularly the ones with time limits.

And for heuristic reviews, to add accessibility in the heuristic reviews, have them reviewed both by usability experts and by accessibility experts. Accessibility experts can go through some wireframes and let you know some quick places to change in the designs.

As for choosing your code frameworks and code libraries, try to find libraries which have some accessibility support. And also code to standards. The WCAG standard is a good one to go by and understand that. You can also use automated accessibility tools to help you check for errors as you code.

So let's go look at some examples of these artifacts. First one, annotated wireframes. This is an example where we lay out not only all the hierarchies and headings, but we also lay out the read order for this design. And at the bottom, if it's a button, colored button that is a link, make sure that you say it's a link. It helps the developer out.

Here's an example by Aiden Tierney. It's a wireframe of a flight booking system on a mobile. So from the wireframe, Aiden's already identified that here are some headings and buttons that we want to make sure that they are coded as headings and buttons. And he even conveyed the disabled state for the number of travelers toggle-- subtract and addition buttons. Also, because he knows accessibility, he knows to tell the developers when the number of travelers is changed, to make sure to announce that to the voiceover to the AT user.

Another example that I picked up this year at CSUN was from Target. Ryan Strunk and Joe Lonsky, they had a presentation about doing a verbal walkthrough. Ryan is an accessibility expert and Joe is the designer.

So Joe, the designer, takes his mock-up and starts telling Ryan, do a verbal walkthrough, starts telling Ryan, hey, I've got a screen with a Back button on the top left, and it has a search field following that. And on the next line there are three tabs. And the middle of the tab, the center of the tab is selected. And before Joe could say some more, Ryan said, well, how do I know which one is selected? And so that kind of clues Joe into well, we probably should have the word selected prepended to the label on the tabs.

So maybe not all of us have a Ryan on the team. But even if you don't have an accessibility specialist to bounce ideas, try doing a blind test. Verbally walk through your design with a colleague. And see if they can understand the information, the sequence of information that you're presenting to them. You probably have missed out certain states and identifiers or information. Or you might have put the information in a different order. That should help fix things before the final launch.

Do I have any more examples? Ah, here. For expert reviews, you could go to your accessibility expert, or even by Jakob Nielsen 10 usability heuristics, 5 out of the 10 are accessibility related. For example, the first one is visibility of system states. It just calls out. It's the equivalent of make sure all your states and hovers and buttons and links can be seen, particularly with keyboard tabbing and mouse hovers.

All right, five, surface layer. The surface layer is the sensory experience created by the finished product. The sensory could be touch, hearing, vision, spoken, whatever.

And artifacts that are involved in this layer are usually design comps and style guides. Design comps helps with consistencies of all the products. It only provides information to help make smart decisions in the future. And it could also be some branding and image pattern library

guidelines.

And let's go look at some examples of them. So design comps could look like what I'm showing you. It could be designed for both a desktop version and the responsive mobile version. And it outlines where the different cards and elements of the site should be laid out, and the positioning and the colors. That was design comps.

So to add accessibility into all these artifacts, in your design comps, make sure you have all the states spelled out, with different color variations, and you pick out the right fonts and recommendation for the fonts. For color palettes, make sure to check for color contrast. And for images, please label them.

Let's look at some examples. Here's a work in progress design comp. What we have is on the top, there is a menu comp, which identifies which element is selected, what's the rollover state for the element, the hover state. And we've also designed it for both a dark background and a light background.

And underneath it, where we have two looking very similar cards, the designer has outlined what appears as a normal view. And on the hover view, there is a slight zoom in the image, and also the underline appears in the link.

An example from Berkeley is their color palette. This I thought was a great color palette. It has same range from vibrant to the classic colors, and has heavy to light. But each of these color combinations have been tested for color contrast. So when Berkeley says put this out, their developers just need to pick from these sets of colors, and they don't even need to worry about color contrast.

For icons-- remember, [INAUDIBLE] icons. OK, my pet peeve. So one day I was playing-- I started playing this game called Endless Legend and I had to go through this tutorial. And as you can read the instructions, it says, "You can toggle the display of the terrain resource by clicking. Click on the buttons located at the bottom right." So there's click, and then there's the little icon. Click something on the button.

And I looked at this, and I'm like, wow, that looks like the front of an igloo or a cave or something kind of curvy, inverted U. I don't know what it is. So I flip on to the next tutorial card, and it shows that same igloo again. And it now says you can zoom in and out for a better overview. And I'm going, well, what's this? I can't zoom at all.

And what happened now, I tried drag, control drag, keyboard arrows. I tried WASD for the keyboard, command plus. Nothing. I tried to look for a magnifying glass-- nothing. So I had to skip it.

And then a few more times down the road, a few more tutorial pages down, I hit another problem. So I started flipping through the tutorials really quickly. And then it suddenly occurred to me that there is a difference between these two igloos. And one of them has a red on the left, and the other one has red on the center. And I thought, and it finally hit me that it's actually the top half of a mouse.

Why is this icon so unfamiliar for me, and why I didn't make that connection earlier? It's because I use a Wacom pen. I don't even have a scroll bar, a scroll wheel to work with. So I had to go look for a mouse.

So moral of the story-- label your icons. Don't assume that everybody uses the same device as you.

All right. So I said I had five layers. I lied. Well, I snuck in a sixth layer, because with the current engineering model, which is continuous delivery, we do a lot of agile products, and the developers need some agile feedback.

And it means essentially it's-- developers nowadays run on sprints, which are very short development cycles of a couple weeks. They focus only on a few prioritized deliverables. And they push out those features continuously.

And so I feel that there is a need to add in accessibility and usability checks while developers are doing these short cycles. So there's sprints. There's limited features. There also are user acceptance tests on these features. And many companies and teams run some user feedback, usually questionnaires to get information from the users. And it feeds back into new features that they develop.

An example of a user acceptance test looks like this. You have the test name, and you have multiple steps to do the test, and you have some expected results field, and the person has to put in pass or fail. And it's a really long list. I'm showing you I think-- this was like one fifth of the page of this particular UAT test.

So how do we add accessibility in there? In sprints, automate. Automate a lot. You can find a

repeated behavior. Automate it in the regression test.

And only save the time you want to spend doing manual tests by time boxing the test. Maybe you say, I have half an hour to do a manual test. Give me something to test, and I won't test the repeated features, just the improved or additional features.

Try to build an accessibility tests for every sprint. And you might not be testing the same thing. You might use this half an hour, whatever, to cover accessibility for separate features.

And for the user acceptance test, run the screen reader test or keyboard test with the UAT script, so the script's already written out. And it's how the other people test anyway, so test it with screen readers. And you might have a UAT test script for accessibility if you're doing a very big project.

And finally, for getting feedback, make sure that if you do get feedback in your help and contact page, prioritize those accessibility requests, if there are any. And periodically run some WCAG evaluation. If you hit a big milestone, that's a time to pull in and schedule maybe an hour or two with your accessibility consultants or experts and run through the WCAG.

So that was how to sustain this continuous delivery model. An example from Yahoo is they actually moved some of the simple tests to the developers. Their Accessibility Lab tells the developers, don't let us find these problems. They're so simple you can check them yourself. And they can use their accessibility experts for the more complex issues.

I think that's all I have. So I hope I've given you some ideas on how usability and accessibility can work together at every level. I'll be happy if you can walk away today with just one thing that you can implement in your workflow. I'm ready for questions.

**PATRICK LOFTUS:** Great. Thank, you Jiatyan. Thank you so much. I'm going to go ahead and get started with the Q&A. Someone is asking Jiatyan, would you agree that we need to have multiple accessibility experts with different skills that are specific to each of the production stages so that they can consult developers specifically at each stage?

**JIATYAN CHEN:** I think that might be too fine-grained. Even though I have presented to you as different levels, usually teams and projects go between these levels a lot. So I don't think we're in a stage where we can chart out each level-- chart out the boundaries between each level that well. I would suggest that your accessibility experts get to know a little bit more about the strategic and the scope layer, as well. Right now I'm seeing a lot of concentration on the skeleton level.

**PATRICK** Great, thank you. And how do you get your development team to buy into the need to test for  
**LOFTUS:** accessibility?

**JIATYAN CHEN:** Get my development team to buy into the needs-- that's probably another topic altogether. For our university, I actually look at the culture of how our teams work and how the university works here, and try to capitalize on what drives your developers.

Your developers might-- for Stanford, particularly, our developers want to be the best. So all I need to go tell them is something's a little bit wrong here, and they jump on it. For other teams, it could be that the boss says so. Or for other people, for other teams, it could be that it might be individual developers who have a personal connection or personal story with somebody with disability, and/or they feel that there is a social responsibility. So it varies.

**PATRICK** So kind of on the same topic, someone's asking, can you talk a little more about how you got  
**LOFTUS:** buy-in for Universal Design for Learning, or UDL, in accessibility at Stanford?

**JIATYAN CHEN:** One at a time. There's not really an overall structure and messaging system for Stanford. So I would go to each department, and we have different conversations.

And some of them are developers. Some of them are communicators, and some faculty and instructional designers. So it really varies. I haven't got total buy-in. I'm still working on it. That's why I'm the program manager. There's a lot of outreach going on.

And while going out there talking to people, I also get information of what they actually need to do their job. And not only do we have to get their buy-in, we also had to provide means for them to do we want them to do.

**PATRICK** Great, thank you. And someone is asking, what are the major issues you see at the college  
**LOFTUS:** level? I'm assuming they mean just for accessibility, or maybe in terms of collaboration between developers and accessibility teams.

**JIATYAN CHEN:** Major issues I see at college level-- teaching content aside, we haven't done a good-- for teaching content, we haven't done a good inventory of what's good or bad. But other main issues I see are developers are not quite testing for keyboard accessibility just yet. It's not built into their development flow. So I'm doing some outreach, as in, here's some quick checkpoints that you can do, quick checks that we can do for the developers, and just hand out flyers.

For content editors, for content authors, I think most of what we're doing is I reach out to the communicators and have them tell their content authors, here are some quick things you can do with headers. Here's how you label all text correctly.

And gradually, there is some kind of a cultural shift. Once people get comfortable with their workflow, they can take in a little more at a time, but slowly. I don't know if I answered the question.

**PATRICK LOFTUS:** I think that was great. Thank you. Someone's asking, do you have any good resources for accessibility training for our developers?

**JIATYAN CHEN:** Good resources for training-- somebody was asking about this on Slack. I don't have any that I can call offhand, no. Usually, I would go to either WebAIM or WAI websites and look around.

**PATRICK LOFTUS:** I think that's definitely a good place to start. Someone else is asking, what is the reasoning behind hiring a vendor to do an accessibility audit and provide a feedback report?

**JIATYAN CHEN:** Could you repeat the question?

**PATRICK LOFTUS:** Sorry. What is the reasoning behind hiring a vendor to do an accessibility audit and provide a feedback report?

**JIATYAN CHEN:** Multiple reasons. You don't particularly need to, but if you have a big project and you've already budgeted in the funds, why not get a vendor to do it? The vendor's report will teach you as well. They can see things in a slightly different way.

But to hire-- let me think a little bit. We sometimes hire out to vendors because it's such an extensive project, such an extensive review. It's the key-- for universities it would be, say, the HR or the time tracking system, or the student information system that has to be completely accessible. And that's when we would budget in money to get a vendor to do a complete audit. For regular websites which have a little bit of wiggle room, we usually do a soft evaluation, and we work on the problems incrementally.

**PATRICK LOFTUS:** Great. Thank you, Jiatyan. This is a pretty good question here. Do you have any advice for bringing up the case of broader usability to our school? We only make content accessible for accommodation requests.

**JIATYAN CHEN:** I would look at some of the situational disabilities. Usually when you fix an accessibility

problem, not only do you make it accessible, you also make your product more inclusive. So if we want to go back to my persona Jill's example, once I put on captions, not only would it benefit people who are deaf. Jill, who has to work in a noisy environment with the kids running around screaming at each other, she can put on headphones and listen to her course.

And you know, taking the Caltrain also makes me realize that a lot of people watch their lynda.com. Or I think that I saw somebody watching football playoffs on their phones in a Caltrain, really noisy environment. But it has captions. So usability gets expanded once you add in accessibility features.

**PATRICK LOFTUS:** Great, thank you. I think we have time for maybe one or two more questions here. Someone is asking, how do you manage programs while always keeping an eye on accessibility? And how do you make sure everyone does their jobs? And how do you think accessibility program managers should get involved in the high-level strategy/decision-making process?

**JIATYAN CHEN:** Wow, that's a long question.

**PATRICK LOFTUS:** [INAUDIBLE]

**JIATYAN CHEN:** How do I keep them all together? Sometimes I wonder that about myself, too. So as a program manager, I try to look further up the chain, and go to meetings where I can effect more change. Like for example, I would have more meetings with communication managers or web managers to advise them on how they can task their developers or their content authors. So most of my meetings are with directors or managers.

Sometimes I do need to dig into the details. For example, I do meet with the instructional designers when they're designing, say, a course, to train the faculty on accessibility. And that I will get to the nitty-gritty details.

Most of the stuff I would-- when developers or content editors ask, I don't spend very detailed time with them. I send them to different links. I have a list of probably more than 200 links in my little database of what's pertinent to their questions, and I just send them over there. The nice thing with working at Stanford is that, as I said, people want to do well, and they really know how to self help. So one or two links to them, and they know where to proceed, and they can go down the rabbit hole.

**PATRICK** Great. Well, thank you, Jiatyan. I think that's about all the time we have left for today. And

**LOFTUS:** thank you to everyone for joining.