1D Designing Online Interviews
April 11, 2022

Tim Baran:

So let's go ahead and start. Welcome to the How to Design an Online Interview Tool session. I'm Tim Baran, the Technology Innovation Program Manager at Pro Bono Net. And our two esteemed presenters today are Simon Rah and Chris Schwartz. Simon is the Senior Technical Project Manager here at Pro Bono Net and manages implementation of projects that include building online tools and resources, like you'll see here. Christopher Schwartz, or Chris Schwartz, as I call him, is an attorney and a community empowerment advisor at LegalServer. He helps legal services providers customize and implement their case management systems.

So we're going to have an opportunity to pause during this presentation and address some questions, so please feel free to put your questions in the chat, and along the way, we'll a pause a few times. I think this is the last session of the day, so we'll save some time at the end to address any questions that we didn't get a chance to address during the presentation. All right, Simon. Take it away.

Simon Rah:

Okay. Thank you, Tim. So this is what we're going to be covering today. After the introduction, I'll describe what the purpose of the presentation is, because I think we need to clarify exactly what we're covering. And then we'll go through the online interviews, the process for actually designing it, and then I'll touch upon some authoring tools that are out there and then end up with the Q&A at the end. But as Tim mentioned, in the middle of the presentation, we will pause two or three places in the middle, so give the audience an opportunity to ask questions and address them during those periods.

Okay. So what is the purpose of this? To clarify, the presentation will talk about what kind of questions should be asked in order to get the result that you want. So it'll talk about the necessary questions for the interview, how to document the questions, and how people are going to flow through the questions in such a way that the developer who's going to create the online interview could take that information and then encode it without having to do a lot of back and forth. What this is not about is how to design it from a wording perspective or a content perspective. This is more of the nuts and bolts of putting together a questionnaire and handing it off to the developers.

Near the end of it, we will present the worksheets which you could actually use to actually document all the information, and we show you step by step how to do that.

Christopher Schwartz:

And during the pandemic, certainly, I think the prevalence of online intake interviews have increased. And we at the Permanent Commission have been advocates for every program to have online intake, if possible. It's such a valuable resource for serving your clients. And I think two of the biggest concerns that people raise with online intakes when they're considering implementing them, the first one is not knowing where to begin. And Simon is going to give some great structure. And I can say from being on the side of being an attorney who's not very wise in the ways of technology, and then being on the other side where, actually, I have to put this stuff together, it's going to be so much easier for you if you have a structured format to put these questions together, like Simon's going to provide us with today.

The other big concern though that I think a lot of organizations have is, "If we turn on online intake, we're going to open the floodgates and we're going to drown." Right? "We're going to have so many applications, we're not going to have any control over them, because once you put something on the internet, it's out of your control." And that's simply not true. And two things I want to say about that is, you still maintain control over when your online intake is available. And by simply putting up a page saying you're no longer accepting online intake applications for the rest of this month, or for the summer, or for this holiday period, you can turn off online intake. You can control that spigot that you might be terrified of.

And the other thing is reversing your perspective and looking at it from an efficiency standpoint, where you think about how many folks come to you for assistance, that you can very quickly tell are not going to make it to a full case plan, to a full case in your case management system, they just need some information about what to do next. Your online intake interview can be designed to do that, to really triage who should get the full way, the full path to full intake, and who could be provided with information that's still going to help them and still going to be beneficial but doesn't require the intervention of one of your valuable and busy case handlers.

Simon Rah:

Right. Thank you, Chris. That, actually, is very helpful to provide the context. Because I want to explain that online interviews could be used in many different ways. And people tend to think of it as a computer screen and then you answer a bunch of questions, but it could be used in mobile phones for questionnaires, or even as a chat bot and so forth. But after the questions are asked, what you could do with the answers are also ... there are many different options as well. You could generate, of course, legal forms that you could then e-file. You could also have result pages that provide some kind recommendations, or further study, or further reading information. Or you could actually have actions that says, "After you fill out this form, an account is automatically created," or an email is sent, or a bunch of things that could be triggered from the interviews.

So what we're going to be covering is the data collection portion, the actual interview itself, how to design it in such a way that we're going to provide these kind of actions at the end of it. We will go through some of the examples. These are the ones that we know of. Of course, there are many others. In fact, during the Rapid Fire presentation, there were three different online interviews that were mentioned. One for LIFT's Family Law Navigator, Reclamo, which is a wage theft system, as well as the security toolkit. Those are all examples of online interviews. I won't go through these right now because of timing, but if we have time near the end, I'd be happy to show you. And of course, you have access to the links, if you want to try them out yourself.

So let's talk about the actual design portion of it. So the most common method of designing a questionnaire is conceptually what people think of as the decision tree or dialogue tree. So you start from the top, you have a question, and then based on the answer, you branch out into two possible questions. And then from those questions, you branch into other questions, and so forth. So here's an example where this is more of a decision tree. And then from that, you can translate them into a dialogue trees, so you have questions instead of these decision points.

There are two main problems with this method. One is that each of the branching is based on the particular question where they branch from. So for example, in the very first branching, whether it's a no-fault or fault divorce, that is based on the specific question that was asked just before it. In real life, though, what you find is that certain questions are asked because of a combination of answers that were previously given. So do you want to ask a question because someone answered that they're married with children who are over the age of 18. That uses a combination of answers to determine whether a question should be asked. So that doesn't quite fit within this dialogue tree model.

A second problem is that because we have these branching, and mathematically speaking, if you imagine each question has a yes/no answer and yes/no gives you different question, by the time you get to the 10th question, you actually have to write up about almost 1,000 questions to consider all the different scenarios of people choosing yes/no for each question. So that gets to be quite unmanageable, as you can imagine. Just by 10 questions, you end up with 1,000 questions that you have to program then.

So an alternative, and in fact, a lot of the online tools that allow you to create these online interviews prefer to do what's called backwards chaining. So instead of actually starting with one question then [inaudible 00:08:40] branching to the next question, and so forth, what you start with is essentially the end results. What is the document that you want generate? What is the result page you want to create? And things like that. Once you have that defined, then that will determine what kind of questions should be asked in order to fill in those holes, the dynamic variables that you need to generate those results.

And then from that, then you could start putting in the variable names, and the calculations involved to get those results. So instead of thinking of it as question by question, you think of it more in terms of the end results, the questions needed for the end result, and then the formulas to actually generate the end result. So those are the steps you usually go through, rather than going through question by question. As is more natural, if I've experienced having live interviews with people, you tend to think in terms of question by question. But when you're designing an interview like this, you want to really think more in terms of the results and then determine the question from the results.

Christopher Schwartz:

And using this methodology will also help you escape the trap of basically putting together an interrogation for someone who is in crisis and looking for legal services, right? Starting with question, after question, after question, doesn't really build a rapport. And when you think about the communities that a lot of us serve, they're keenly aware of being pumped for information, of being added to mailing lists, of having to reveal personal information when you just have question-to-question format. These interviews work better if there's a sort of give and take, where you're asking for some information. Maybe tell us what zip code you're in and we'll take it from there.

And then you're providing a little bit of information. "You're in a zip code that has a lot of rent regulated housing. Would you like to see a GHCR information page on rent regulated housing?" You can provide information, you can provide that in a conversational methodology without just having a straight, "Name, address, income, how many kids do you have in your household?" Which is I think what a lot of lawyers are prone to do, because that's how our user-facing intakes are set up. So using Simon's methodology and knowing where you want to go will let you not only reconstruct the questions in a logical framework that works and avoid 10,000 questions, but it'll also let you have a bit more of a rapport building experience with those you're trying to pull information from.

Simon Rah:

Right. Thank you. And this also solves one of the biggest problems. Well, the two issues with the forward chaining. One is that you could actually reduce the number of questions you ask, because you know that the questions you're asking are specific to the target results. But also, it does mean that you don't have to duplicate all the question because of branching. Secondly, each question could be set up so that it's based on any previous answer that was given. So a particular question could be shown because someone said that they were married with two plus children, and all the children are over the age of 18. You could have compound complex rules like that to determine how each of the questions will be shown or not shown to each person.

Okay, so I want to kind of pause here to see if there were any questions in terms of the logic, or the purpose of using this method.

Tim Baran:

So Simon, no questions in the chat yet.

Simon Rah:

Okay.

Tim Baran:

But I do have a quick question, just based on some of the projects that I've worked on with these. To your point and Chris's point, there are the default is to be as comprehensive as possible. You want to ask every single question to address every possible iteration of something that you're trying to resolve. And that becomes very, very large, very, very quickly. What do you tell a program partner that wants to include as many questions as possible to be as comprehensive as possible, to sort of scale that back? That's question one. And the other part is, do you recommend starting with a really scaled back version, and then iterating on that as you go on, just based on data, or not, or try to come with a really top of the line product right from the get-go?

Simon Rah:

Yes. In fact, that's one of the most common errors people make, that they want to ask as many questions as possible, with the understanding that maybe down the line, you might want to use that data for other purposes. That is usually not a good way to go, because you're asking a lot of questions that are not currently needed. It also means that the users going through the questionnaire are more like to drop off during the middle of the questionnaire or interview. And you want people to go through the full interview as much as possible. So you want to keep them as short as possible while addressing the need of the results, so that should be your target.

Now, if you do want to elaborate and ask for contact information just in case you want to reach out to them afterwards, you could do so. But what I usually ask for every question that's asked is, why is this question necessary? What is this question supposed to gain us by getting the answers? So having that kind of dialogue with the subject matter experts, we could narrow down the questions that's needed to actually create the interview that we want.

Tim Baran:

Awesome.

Simon Rah:

Okay. So I will go through a very quick example of how to use this process. So let's say I have a high-level objective. My goal is to do an intake eligibility questionnaire, and I want to know if people who do the questionnaire will be eligible to receive my services in Simon's Awesome Legal Aid. And if they don't, they're not eligible, we want to give them some information about why they're not eligible, or allow them to go elsewhere for additional information.

So I define my result as having these four particular potential results. One is, they're eligible because they live in New York. The issue that they have is housing issue, and their income is less than $1,000 a month. The other results are when they're not eligible but we want to provide them a bit more information about why they're not eligible or additional to help that they could get. So second one is if they make more than $1,000, we warn them that you need to make less than $1,000. Third result is, if it's not a housing related issue, you could go to lawhelpnewyork.org to find information about other legal areas that could help them, and other services that could provide them the help that they need. And lastly, if the person's not a New York resident, we simply say, "Sorry, this not interview is not for you."

Christopher Schwartz:

I would also make a pitch for ... and I know we said we weren't going to talk too much about language, but I think as sort of from a design perspective, or as a design principle avoiding, "Sorry, you don't qualify," kind of statements. There are other ways to say the same thing, right? I think a good design principle would be, I want everyone to walk away from spending their time on this interview with something. So when Simon is redirecting to LawHelp, he's providing a reference, right? He's also providing information that where they are currently is not the right place for them.

And I think by saying, "There's no reason to continue down this path," you're providing a service. You're giving someone information about wasting their time. Their time is as valuable as our time sometimes, and the efficiencies that we're presenting them should be recognized. And also, when you use language like, "Sorry, you don't qualify," or, "I'm sorry for you not being in our catchment area," again, dealing with people with crisis, the last thing they need to hear is, "I've already failed and I'm just applying for services." So just think about what you're providing as an alternative, if a person really isn't eligible for your services.

Simon Rah:

And as you can see, for each of the four potential results that they could see, we have rules written out. So the first text, "Congratulations, you're eligible," will only show up if they live in New York, has a housing-related issue, and the income is less than $1,000. So each of these results need to have some kind of rule applied to them. Or it could be a sentence that says, "Hello, welcome to…" whatever, in which case, it's something that's shown to everyone. But each text or screen or document that you're going to generate has to have something that you want to display then. That's important for step two, because for each of the rules, you want to determine what the data points are, what is the information you could get from the user in order to calculate, run through the rules to see if it's true or false?

So, in our example, we have three different pieces of information we need. We need to know where they live, and what kind of legal issue they're dealing with, and what their monthly income is. From that, we could extrapolate the question, of course, and this is the most straightforward we could ask. Where do you live? What is your legal status? And what is your monthly income? But of course, we want to massage that a bit, so they're more user-friendly and clear. So that instead of just asking, "Where do you live," you might want give them an option of a particular state, a list of states that they could choose from, or it might even turn it into, "Do you live in New York, versus not," that kind of thing, depending on how you want to board it.

So that's where the refining comes in. You take the questions and then refine it so that it's more user-friendly and clearer. And as a general rule, all the questions, you want to try to keep it as multiple choice questions, if at all possible. That's a lot easier to answer than if they have to type something in. Not only is it more useful for the users, but also, it's a lot easier than to do with formulas. Because remember, we have to take these answers and we have to put it into some formulas, so that we could test it against some rules that we have.

So that's where the refining comes in. Once you refine the question and you say that, here are the different values that they could choose from, you also need to then define the variable name for each of the questions. Because each answer is going to be stored as the variable name. And those variable names are very important, because later on, when the developer programmer puts these questions in, they have to reference each of the answers by the variable names.

The other part that you're seeing here is the Showif rule, and then the contextual help. For each question, you could have a rule that says, "Show this question only if a condition exists." So in this case, for example, we said, ask about the legal issue only if they live in New York, if they say yes to New York. So that way, you don't waste the user's time. So you don't have to ask any further questions beyond, "Where do you live," if they don't happen to live in New York. And they could just be given one question rather than three potential questions.

Contextual help is also helpful, because as a general rule, you want your questions to be as short as possible so it's easy to read. One, potentially two sentences. And any additional clarifications of the questions should be included in a contextual help section. So for example, let's say you're defining what New York City means, what boroughs does that cover? Or you're defining what housing legal covers, so you might want to put that into contextual help. This is very helpful, because some of the people will understand the question without having to get contextual help, while others will need to read the more detailed, contextual help to clarify the questions. It provides the right balance of being able to go through the questions quickly for some, and for others, taking their time to really understand each of the questions that are being asked.

So with the questions and the variables, now you could then reword your rules into these variable names. So residence equals yes, legal issues equals housing, income is less than $1,000. This kind of a pseudo mathematical formula is important when we talk about programming. Because there's some things that seem obvious, things like they're in New York City, which are not necessarily clear to a programmer. So for example, if you ask, "Please put your address," and then from that address, we have to extrapolate whether they live in New York City. It's not always clear to the programmer, whether they should take the name of the city and then determine whether it says New York, and Brooklyn, or Bronx or so forth, or they only look for the word New York. So those kind of rules have to be very specific, in preparation for the programmer.

Yeah. So I know I'm starting to get a little technical here, so I want to pause a little bit to see if anyone has any questions. So I just-

Christopher Schwartz:

We have a good question in the chat now.

Simon Rah:

Mm-hmm.

Tim Baran:

Yeah. You can go ahead, Chris or Simon, if you want to do that.

Christopher Schwartz:

Sure. I can take this one. And then Simon, you can add your layer of understanding as well. From John Antal-

Simon Rah:

Could you actually-

Christopher Schwartz:

Mm-hmm?

Simon Rah:

Yeah. Oh, sorry. Could you read out the question? Because I can't see the-

Tim Baran:

Yeah.

Christopher Schwartz:

Yeah. Sure, sure. So from John Antal, for questions that require additional clarification, how do you provide an option to get this clarification? For example, does my monthly income include social security, or just my W2 employment, to clarify where is the contextual help located?

So since I read it and I'm going to have the more simplistic answer than Simon, I would say it's a question about the tool that you're using to build the online interview. Right? I know that, for example, A2J Author provides you with the ability to pop out a text box, that if someone has a question about one of the phrases, they'll see a hyperlink under that phrase, "What does income entail," click on income, and then that pop out text box can say something like, "Income includes, for our purposes, this, this and that, but not this, this and that." For something like Legal Server, you can have hover over text, or you can use help text as well to give you something similar. If you have a question about income and you hover over it, you can get an explanation from that. And then you always have simple things, like just hyperlinking to a glossary of terms, if you will.

Simon Rah:

Right. And what I'm showing is an example of another way of doing contextual help. So contextual help usually means it's kind of hidden until someone needs additional help. So there's usually a link somewhere near the question, where if you click on it, it'll give you more information, of what it means to be inside the US, for example. In this case, what does it mean to have immigration court case? So it explain how to answer this. So that's what we mean by contextual help.

But another way you can do contextual help, especially for questions that are more complicated, is you could ask a very simple question. In our case, we were talking about monthly income. So you could say, "What is your take home amount?" And if you say, "I don't know," or, "I'm not sure," then you could have a follow up question that breaks it down into multiple parts. Like, "What do you bring home every month? What does your spouse bring home every month?" And you could kind of ask them separately as follow up questions, and use that as a way of doing contextual help.

Christopher Schwartz:

And I would say you should always have a panic button as well, right? Not just for questions that confuse someone, but also for any sort of technical difficulty. You don't want someone to drop off of your interview and potentially not get help because they can't finish your interview. So you should always make alternatives available, whether it's information about your walk-in clinic, or the main office number. That should never be very far from any page or any step of the interview that the person is on.

Simon Rah:

Right. Okay.

Tim Baran:

Hey, Simon. I know you have a lot to cover. There are a couple of questions that I have, but I'd say just, you could go ahead and continue, and then we could address it towards the end.

Simon Rah:

Okay. So yeah, the second to last part is, I want to introduce the worksheet, because this is how you document the different questions and the rules and variables in one place, so that the program or developer will be able to see exactly what needs to be coded and how. So what I showed you so far is essentially the steps that's taken, where you first talk about the outputs, you think about the output. You think about the rules for when to display the output, and then you come up with the questions based on the rules that needs to be calculated. And then you kind of refine the questions and the rules and the outputs again, and again, and again, until you get it just right.

So the worksheet allows you to document all of that kind of together. So this link that's shown here on top is a linked way Google Worksheet that you could use, and it has several tabs of examples used for intake, or for if you're doing document generation, or a triage kind of online interview. It'll give you examples of how to organize that as output, output, rule, question, and display rule, so forth. So I'm going to go through this part pretty quickly. And I'll show you the actual sheet after this.

So on the far right, we have the output. So this is where you put in the actual results that you want to display, on the screen, or in the document. You put the rules in kind of pseudo-English, what the rules are. Once you're done with that, then you could determine what the questions are going to be, the title of the question, the options. And this is where you specify the variables. Each of the options get the variable names, because that's usually how it's coded. And then from that, you could then refine this a bit, that says, "Okay, show this second question only if they say that they live in New York City." If they don't, then you could skip this question and go to the next question. So that's what the display, if rule is.

And here is an example where you have a more complex rule, where if the residence is yes, and the legal issue is housing, only then display the third question. So you could put in more complex rules about when to show each of the questions. Once you have your variables, residence, legal issue and income, you could then put those into as a formula that the programmer could then code in. So it's clear what it means to live in New York City, means that they chose residence Y for the answer. So this will make it very clear so there's no misunderstanding. Okay.

So I'll go here and actually show you the sheet. This is the sheet. So this is the link that we provided, we shared on the presentation. The first sheet is blank, so this is what you could make a copy of and then use. NYSTech tab is the one that I just showed you with the example intake form that we were just going through. Here's an example, something more complex that says this is the intake form Word document that's going to be generated with these variable names from these questions. Here's another example of an intake. And this is a triage example or a public charge. This is for determining what kind of public benefits you're eligible for when you're an immigrant. So it has some examples of how to organize the questions and the outputs so you're [inaudible 00:29:30] with.

I know it's difficult to see, but I just want to just let you know what's here. So each of the columns gives you the specific information that you need to put in, and then this could be as long as you want, and just keep on adding more and more rows for each of the questions that you're adding in. So let me pause here, see if anyone has any questions about how to use this.

Christopher Schwartz:

Let me put another tool into chat while we are talking. Many of you might be aware of this, but I think it's important to raise every now and again, a tool that was put together by different LSC partner programs that we had. And some of our colleagues who were also part of this tech committee at one point in time called WriteClearly. And WriteClearly is a website, it's a little applet that you could put on your browser bar. And as you're creating the content, the actual copy for your site, it's going to allow you to analyze the sort of grade level of the information that you're conveying.

I think the site said, I was looking at it earlier, that most Americans, the average American level, the English first language reading level is an eighth grade level. And obviously, we should not be writing at an eighth grade level. We're dealing with marginalized communities, we're dealing with folks who have English as a second language. We should think more, I think, between third to sixth grade level, for many of our questions where we can, many of the information that we can convey where we can. So that tool will give you a general grade point level of the copy that you're producing.

Simon Rah:

And I guess, one more tip for when you're writing up questions is try not to use compound questions. So for example, you don't want to say, "Do you live in New York, and do you have issues with housing in New York?" But you don't want to ask that as a single question. You almost always want to ask a question that deals with one specific topic. That way, if anything, you might want to break it up into multiple questions, if you want to dive into more specifics. But you should never have compound questions if at all possible, because that makes it very complex in terms of coding.

Tim Baran:

So I'd like to address this question to both you and Chris and Simon. When you were building tools like this, it's never in a vacuum. It's always a collaborative effort, right? You have several stakeholders, you have several partners that are working together on a tool like this. You're starting with questions. How do you engage a co-design or human-centered design process in creating a program like this, before you even go to the worksheet? If, Chris, I could start with you.

Christopher Schwartz:

Sure.

Simon Rah:

Yeah. Go ahead.

Christopher Schwartz:

And Tim, you might need to refine that question just to make sure that I'm getting the right thrust. But I mean, honestly, I'm looking at all the folks who are joining us today, and I see several folks who I've designed online intake interviews with. And I think the most successful starting point is to think about who's going to be answering the online intake interviews, and what your goals are. Are your goals to have a full case presented and ready to go in front of an attorney? Or is it something more like, this is going to be another avenue for us to provide assistance, so we want to do something that looks more like minimum eligibility and a little bit of an idea of what kind of legal problem you're having, in addition to providing information.

And I think the human-centric part of that is realizing that it shouldn't just be a wall of text, or a bunch of frequently asked questions that you're throwing at people. You shouldn't be populating your interview with statutory information. You're having a conversation, and you're trying to build a rapport. And I keep coming back to that, because it's so important to me. You're exchanging information, even if it's not a live version of you or one of your case handlers. You're exchanging information with the person who is going through the online intake interview. I think you'll get a more successful interview that way with more utility, and you'll find that more people finish your interviews, when you take it from perspective.

Simon Rah:

Right. And one of the reasons that I like to use this worksheet, I'm presenting this worksheet is because when I work with a subject matter expert their coming [inaudible 00:34:33] to the interview, this worksheet clearly indicate which part of information is needed for the online interview to be created. So they need to define what output they want in and write it out, even if it's normal English, and they need to write out the questions and the options available. That part is something that the subject matter experts could definitely come up with and refine as needed. If they're comfortable with it, then they could also put in the variable names, and then they'll convert it into a formula as needed.

But that part, that's where the gray area is. Typically, I do step in at that point, I will put in the variable names, and I would convert the written rules into a formula, because I know that's how it needs to go into whatever tool I'm going to be using. So that's where, I guess, the handshake happens. The output and the initial set of questions are done by the subject matter experts, and then I would come in with the variables and the formulas with the output rules. And then, of course, code it into the actual tool that we're using.

Christopher Schwartz:

And using a shared document, just like this one, is a great way to do that, where you can both go in, work asynchronously, but still see the progress of the other party.

Simon Rah:

Right, exactly.

Tim Baran:

Okay, great. So Simon, you can go ahead and continue. We have about 12 minutes or so left.

Simon Rah:

Okay. So I just want to end with the giving a list of authoring tools that Chris and I are aware of. I've used most of these, not all of them. And this kind of explained what some tools are better for than others. Personally, I like Afterpattern/Docassemble, because I think it has a good balance of ease of use, for even non-technical people, while giving enough flexibility to really come up with the complex interviews, if you want to. It is also free to use. But in Pro Bono Net, I've actually used multiple tools depending on the needs of the project. In fact, some were also built from the ground up, because of some of the limitations of the tools. So I won't go through all of them, but Chris, if you could say a little bit about the Legal Server Authoring, because I'm actually not familiar with that.

Christopher Schwartz:

It's okay. I don't think they're paying me for ... No, no. Yeah, I've done most of my authoring work in Legal Server, because that's the system I'm familiar with. But if your program uses Legal Server, you're not forced to build your online intake in Legal Server, but it works, because it's integrated with your case management system already. And other programs could be built to do that as well, but it's a little bit easier, especially if you're familiar with building forms and processes for regular intake and Legal Server. The online intake works that way too, so you'll be using a familiar tool. And that's probably, in some regards, more important.

I mean, I think that we're approaching this from the perspective of, you will be working with someone as smart as Simon to do all the actual technical work. And that's not always the case, right? Sometimes you're not going to be able to hire a consultant, or hire out, or you don't have an IT person on staff. So think about not only the initial build, but maintain it. What's going to happen when some of your eligibility criteria change and you have to go in and change the interview? You're going to have to go back to the same tech person who did the interview for you in the first place? Or would it make sense to, if you have hired someone to build the interview and one of any of these engines, would it make sense to maybe task one of your staff members with getting up to speed, so that the next time they can go into a Docassemble interview and change your eligibility criteria as well? Because as we use these tech tools, becoming aware of all the time and effort that's put into them, and the need for that expertise in-house, I think is important as well.

Simon Rah:

Also bear in mind, some of these tools are very user-friendly, like A2J, like Afterpattern, somewhat Docassemble. As long as you're not doing something too complex, you can use the authoring tool as is to build it without having a program or a technical person involved. If you could do a SurveyMonkey on the survey, then those other kind of tools are similar to that. So it's not unreachable for just a subject matter expert to create the full thing themselves. It does get a little bit more complex, especially when you're dealing with several hundred questions sometimes, but don't think that you must have some technical person in order to actually create the online interview.

Christopher Schwartz:

Yeah. And I think that raises an interesting point, right? If you're the only person, if you're the subject matter expert and the erstwhile computer expert as well, because you're going to be designing the same thing, you should still have a design document. You should still use something to put together, you should still use Simon's form. The utility of that is not only going to focus what you're doing, and the work that you're doing, but that transmission of knowledge to the next person. Unless you think that you're going to live forever and be at the same organization forever, is going to be super important, right? That transmission of what you did and why you did it really comes through when you have a design form like this.

Simon Rah:

So that's it. I just want to end with just showing you the examples I previously showed. These are all the different authoring tools that we use for these different sites. So I think we're ready for questions. And we have contact information at the end, if you want contact us at a later time for additional information, we'd be more than happy to answer any questions afterwards. But are there any questions then?

Tim Baran:

Hey, so Simon, you had mentioned that since we have another five minutes after or so, you mentioned that if we had time at the end, you would just take a look at maybe a tool out there, or maybe one or two tools, just to give an example. But before you go there, Chris, he hit on this a little bit, but when you're building and when you're done building ... Well, you're never really done building, right? So how do you keep up with the maintenance and the build out of these tools, the sustainability of these tools? I know that there are tools like JIRA and Asana that are project management tools where you can keep putting in tasks, "Oh, this is not working," or, "We want to change this," or, "We want to change that." But I know there just needs to be a system in place for continuing to maintain and improve this product.

Simon Rah:

Yes. Not just improving, but the laws change as well, or the rules for eligibility changes over time. So you're forced to make changes to the interview. So really, in order to do that, you have to have it documented and format that's easily readable, because you're not going to be able to go to the authoring tool and understand where all the rules are set. So you need to write it up like this so that a subject matter expert or a technical person could read it and get the same amount of information from it. So this is vital if you want to do additional updates in the future. And as Chris mentioned, it's especially vital if you're changing staff.

Tim Baran:

So Simon, yeah ... Oh, I think you froze there for a second.

Christopher Schwartz:

He looks okay to me.

Tim Baran:

Oh, okay. Okay. Simon, so yeah, if you could just show us really quickly, another five minutes that we have left, example of what you were talking about. Maybe that would give us some context.

Simon Rah:

Right. So the Immi site, for example, is a triage example. This interview, actually, is almost 300 questions, a little less than 300 questions. Because of all the branching, the typical user will go through only about 40 or 50 questions. But this provides a triage of what kind of immigration options are available to you based on your current situation. So it'll ask you a bunch of questions and so forth, and then you end up with a result that looks kind of like this, that says, potentially, you have a U visa option. It'll, based on your answer, give you additional information, as well as next steps. And you could also search for a legal organization to provide help, for example. So this is an example of a triage system.

The Risk Detector, I think was mentioned, in the rapid fire one. This is from Neota Logic. Oops, have to reopen that. This, I believe, is also a triage kind of question, but you guys are probably familiar with these kind of questionnaires. I believe this one was created with Neota Logic. Another one is the Reclamo one that was also in the rapid fire tech. Actually, this one requires a login, so I'm not going to show you that. But I believe another one, this is something that some of you may be familiar with, the LHI HotDocs interviews looks like this. Go for it. Also, and I like A2J interviews, at the more of a single question design, where it asks you one question at a time usually, like so. Oops.

Even though they all look very different, the design of the question and the rules of when to show them, how to branch, and the rules of the result pages are all more or less the same. So that's why it's using this method to create or design the interview, will allow you to use it for pretty much any of the tools that we mentioned. It should apply, if not tool specific.

Christopher Schwartz:

And we have some questions in the chat that I will call out. Pat Malone asks, "How do you know when an interview will be complex?" And that's a deep and existential question sometimes, right? The interview is up to you, honestly. It's a matter of what you want the output to be, and what you want the value that you're providing to the person who goes through the interview to receive. I like focus group testing.

And I think that ties into an answer to another question. How do you know if you got the logic right? You test. If you're the person building it, you probably test until you can't stand it anymore. And you ask some colleagues to test. And most vitally, if you can, if you have a client population who you can pick one or two former clients, make sure there's no vested interest in them providing you with anything. But if you can get them together for a focus group and say, "We've got this new tool. You've had some issues in the same area in the past, would you mind running through this interview for us?" You'll find those breakdowns of logic. Use it until someone breaks it, and then try to fix it, is what I would recommend.

Simon Rah:

And I would add, simply the act of coding your ideas, the rules and the questions, and so forth. The fact that you have to kind of write it out as a formula is a good exercise to determine whether what you're asking for is too complex, or you're not asking the right questions. But that's a good way to check if the rules could be simplified by just re-asking the question a different way.

Christopher Schwartz:

Do you want to go through the other questions in chat, Tim?

Tim Baran:

Yeah, I was going to say another question that just came in. What might a panic button look like for someone who's having difficulties completing an intake form? And, of course, I think you talked about this a little bit.

Christopher Schwartz:

Yeah. It could be something like a button that you program in, "Are you having trouble? Click here." And it could open up something like a no reply email address that they can put some contact information in. I think I would prefer a link that leads to either a page, or a popup that says, "If you're having problems completing our intake form, please give our hotline a call, please call our front desk, please come to walk-in intake, please come to our help desk." Because you don't want to strand someone who has a court date tomorrow with an online intake that they can't finish. And that could just be a hyperlink, it could be a button, it could be a pop up.

Tim Baran:

And I think-Yeah, go ahead. Go ahead. So Simon I'm going to give you the last word. We have about a minute left, just to wrap up, and then we'll call it a wrap.

Simon Rah:

I would say, don't be afraid of starting this. As I showed you in previous screen, there are free tools out there that you could try out, especially some of the easier ones. So try out just creating a set of questions, and see how easy it is to then display it on the web. And then, be disciplined when you're actually creating a real interview. Make sure you have the outputs, the questions, and the variables in the formulas. Only when you have them clearly defined will your interview work as expected.

Christopher Schwartz:

And for the last word, I think I'll take Pat Malone's question, because I think it's a great one, which is ... or it's a comment. We learned the calculating rates and accruals and spending. They're very complex to do on the back end. And I think what I would say there is, do you need that when you're providing this online intake, when you're triaging? Or do you need rather to know that this person is in this zip code, makes under this amount of money, and is having problems in this very broad area that they can describe of legal services? You can get an online intake interview that is not yet ready, it's not yet ripe to place with an attorney. And then you can have someone follow up and really go through, "Here are the accrual rules that I need to apply to this case."

Christopher Schwartz:

Some of these online intake interviews can be about getting the case before the right person, even if they're not ready to just pick up and litigate. Online intake is very valuable for that. And you think about all of the cases, I think when I was in the field, we tried to estimate how long it would take us just to get through your name, your address, your zip code, how much you make. That could be anywhere between five and 10 minutes. So if you're sieving your advocates between five and 10 minutes, by taking in very, very sparse eligibility information and then letting them do the more advanced technical work, your online intake is doing what it's supposed to do. And it's potentially screening out people, and it's saving your advocates time.

Tim Baran:

Awesome. Well, that's a great way to end this program. Thank you, Simon. Thank you, Chris. And thanks everyone for spending your late afternoon with us. Remember to check out tomorrow's program. It's jam packed with some really, really great programming. So take care, and have a great afternoon.

Christopher Schwartz:

And thank you, Tim.

Simon Rah:

Thank you for your attention.