**Globally Speaking**

**Podcast 74 Transcript**

**Tell Me About Internationalization!**

**Featuring Adam Asnes of Lingoport**

Michael I’m Michael Stevens.

Renato I’m Renato Beninatto.

Michael And today on Globally Speaking, we have a guest on a topic that we have not talked much about to date.

Renato Yes. Have you ever seen the acronym i18n?

Michael I’ve seen it around.

Renato What is it, Michael, for you?

Michael It’s internationalization. It’s about making things international, which gets confusing, right? Where is the line of internationalization and localization, and how do these pieces fit together? Sometimes, if people are not clear with the definitions of these terms, they just get everybody confused at the cocktail party.

Renato Absolutely. So, we are talking to one of the biggest experts on this topic today, and let’s hear his story!

Adam Hi, this is Adam Asnes, I’m the CEO/Founder of Lingoport, and we are focused on software development and localization together, with products and services so that your software works well in any language—that is, spoken language or locale—around the world.

 If you think of software, and as you interact with it, there’s a lot of logic that you might take for granted if you’re just, say, an English user. But, let’s say you’re using another language—take French for instance, and you’re from France—you’re going to have all kinds of different changes—not only the language, but even many of the formats like number formats, date formats; the notion of addresses will have differences.

 A combination of software and service helps companies manage that as well as keep up with the pace of change as they add new features to their products. And many of our customers are very well-known global technology companies that you’ve heard of, and they rely on us because often they have a lot of products that have to be presented in multiple languages, and we help them scale that effort.

Renato We talk a lot about localization, which is something that happens after the product is ready. You usually come and talk to software companies and product developers way before the translation and localization happens. Tell us a little bit about that conversation and what is that service, that preparation service that you do, which we call internationalization.

Adam A lot of times, we get customers that are already, you know, doing some level of localization, but they know they don’t have it right. Or, they know they’re not scaling well and dealing with issues time and time again. So, they’re sort of repeatedly banging their heads against a wall.

 But, assuming they’ve never internationalized, companies will often come to us first and foremost based on a business condition and a technical condition. And the business conditions are usually anything from “we’re a global company and we just bought another company, and that company’s products are really US-centric; they only present in one language and one format, and yet we’re a financial company,” or, “we’re some kind of a social network company and we are looking for customers all over the world, so we need you to adapt this software.” And there’s usually a big scary date in front of that.

 Another way that this happens is a company has a strategic objective. So, a company says, “you know, we are selling around the world, but we really just do it with our US English product right now. Maybe we can store Unicode characters in our database, so we can deal with worldwide customer data, so we’re really an elegant application working around the world.”

 And so, there’s a strategic objective. That, too, has a date because, you know, if you’re just going to have a plain objective without a deadline, it’s probably not going to get done and its budget is going to get used for something else.

 And then, finally, the last kind, which is my least favorite kind: “We have a customer in XYZ country and they want to buy, and we need to get this done now.” They tend to be the most volatile and it tends to impact the company the least.

Michael Urgency without strategy often happens in that moment.

Adam Right, yep. And people don’t necessarily realize what they’re taking on, what it’s going to cost, how it’s going to change their organization, how they’re going to have to change their outlook. So, that third one, while it has a lot of that urgency, the deal can go away—often it does.

 You also asked about what concerns those companies have, and often the first concern is, you know, how long is this going to take? How much is this going to cost? And by the way, it’s never cheap. And how am I going to transfer that information? Am I throwing this effort over the wall? How many of my own resources that are focused on other features, how is that going to take away?

 Now, the last part, the technical part of how does this happen, what’s involved, what are the technologies, we have really good answers for that, but typically those are not the drivers of that kind of decision, although it certainly has an effect on who somebody is going to choose to do the work.

Michael So there’s both the upfront cost of investment, but then also the internal resources that it may take for them to enter into this. Which one of those factors do you find companies focus on more?

Adam You know, I think it really depends on the company. I’m strictly speaking of Lingoport’s services. So, companies can hire our company to internationalize an application for them. I would say that the most popular route for that is that the company puts one or two people on the project along with our teams, and so while they’re not going to have their entire development team focus on internationalization for six months, they might have one or two or several key team members interface with our team members, and so that way there’s a lot of speed in the implementation.

 You know, any time you use a company that’s coming in from the outside, somebody new is going to have to learn how your application works. And we have a lot of methodologies to do that, but the best is really to have somebody on the customer side that understands it.

 My druthers are to have some key people on the client side. The more people on the customer side that are involved, the more likely it is that there’ll be some emergency du jour that kind of pulls the developers away. Something comes up, and everybody’s hair is on fire, and people get distracted, whereas a dedicated i18n team will just keep moving because that’s their emergency. I mean, that’s where they’re focused.

 Some companies will say, “We need to be entirely focused on one objective. We’ll answer your questions, but we really want you guys to work on this as a turnkey.”

 I will say that, you know, there’s often a blend from, you know, “help us do it” to “you do it all” that typically happens in between. We do use a lot of technology to make that happen, including our own products, of course. We use all kinds of dashboards and static analysis, which is a more technical term, but ways of measuring progress and things that are changed, that are built into the continuous engineering and continuous deployment.

Renato You mention technology. Lingoport has a tool, a code-checker, something that automates part of this process of avoiding internationalization mistakes. What is the role of machine learning and artificial intelligence in this process, and has this improved or impacted in any way the way internationalization is done?

Adam Okay. Specific to machine learning, number one is scoping the project. Nobody is going to write a blank check for whatever project, whether internally being run or against us. There is the scenario where they haven’t internationalized before, but the more likely in the product side is there’s ongoing development. Development teams never sit on their hands.

 So, even if we’re internationalizing for the first time, there’s going to be a team that’s responsible for that internationalization, but there’s also going to be in parallel teams that work on new features. And so, you’ve got to mix the two, so you want to know that all the teams are using that analysis to make sure they’re not introducing internationalization issues to the code. And that kind of never ends; you don’t internationalize once.

 Now, specifically on machine learning and artificial intelligence, machine learning being sort of like a subset or its own branch of artificial intelligence, but, you know it’s sort of like AI light, we do use machine learning in the scanning capabilities of one of our products which is called Globalyzer. And that analyzes code either at the repository level or right in the developers’ IDE with the second one being sort of better. You want to find issues as they’re being worked on on the developer desktop.

 It takes a little bit of time to train. So, every programming language has its own unique internationalization set of issues and way of handling things. Say, you’re in Java and you use a calendar class, but you didn’t pass the locale selection to the calendar class. That’s sort of a complex situation that you want to find. So, you can use, first of all, rules that are built into our products that would identify an issue like this. JavaScript, for instance, will have embedded strings, which are a problem, because if they’re embedded, the translator can’t get to them, or concatenated strings. Also, JavaScript may use a string as a variable.

 So, you don’t want to translate a string that’s being used as a variable. You’ll break the code. So, you use ways of detecting the difference between the two, and the basic rulesets within Globalyzer will start to handle that. But you can also tune that using a whole machine learning capability in our products, or by writing a regular expression (but it’s kind of more fun to do it with machine learning).

 Where you go through the results—you’ve got to spend a few hours doing this. Let’s say you have half a million lines of code and, you know, Globalyzer finds 20,000 issues and you want to pare that down. First of all, you probably wouldn’t write against the entire repository because, you know, that’s a headache; you only maybe need the initial scoping.

 But, you can pare that down by going through and going, you know, “true/false; true/false,” and the machine starts to learn from that. Again, I’ll emphasize that, you know, if you’re only going to click five issues, you’re probably not going to get a machine learning corpus to take advantage of that, but if you spend a few hours doing that, you’re going to get some quality results that will be valuable over the source of scanning in multiple cases.

 So, yes, there is machine learning there. It’s used to make it very simple, to get better and better automated results as you scan code more and more.

Michael How often are the results of what the machine learning shows and demonstrates fed back to developers working on new features?

Adam So, you have this concept of a sharable rules engine that is sharable based on the account, not necessarily on all of our product users. Those sharable controls—we call them rulesets—are both hosted on a Globalyzer server which can be either installed on premise or hosted. The way that developers work is generally on a repository, but we also made those rulesets available on a repository basis, so developers can really be working locally, can fine tune that stuff and reflect to their own groups or to the company account at large.

 Managing those controls is actually really, really important because they’re all going to be customized based on the application and, sometimes, even the group, and you are going to want to collaborate with developers on that, not just make something monolithic.

Renato The process is very developer-centric, right?

Adam Yep.

Renato How does this impact a localizer, a localization manager, a translator, in their day-to-day lives? How can they identify a problem that goes upstream to the development phase?

Adam It is a basic product development principle that the earlier you catch a defect that’s going to affect some measurement of quality, the cheaper it is to fix. So, again, let’s compare localization to the development process. When you are checking for coding quality, it’s really commonly accepted among developers to find that really early. So as a developer works, either as they’re writing the code, they find the process; as they submit what’s called a pull request, or submit their work back to the repository; or against the entire repository, there is some automated process that checks for this quality that they’re looking for.

 So, if we’re finding that early on in the process—like, let’s say a developer writes a hundred lines of code, which is, you know, not unusual—if they can find that issue right as they’re writing it, it’s really fast to detect. So, you know, if the developer had a bad day and they have three issues that they have to fix, boy, they’re going to fix that in like five minutes, okay?

 If those issues are found later on, during QA or where, you know, they might pseudo-localize or they might have gone all the way to localization, now the testing group has to log a bug. Let’s say there’s a queue of bugs. First of all, some manager or product manager has to assign those bugs. Then, the developer goes, “I don’t know where this is in the code; I’ve got to now search through. I have an inkling of where this might be, but…”

 Many of our customers have, within one product, say, 150 repositories that they have to look in, and unless they’re working in that kind of code and automatically recognize where it is, it’s not a simple, you know, “this equals that” for them. So, they have to go back, figure out what it is and what the problem is.

 So, there’s a lot of human element in that process, and it is no surprise that, you know, you have that gap between software development and localization when these things happen, because the developer has other things they have to do, and they have to go back and search for all this. I mean, this is not what they went to college for or what they learned development for. I mean, this is not a fun project for a development team to do. So, finding it early really streamlines all that.

 I’m also going to say, the second part—and this is maybe just as prevalent for localizers and developers in their interface—is localizers are depending upon developers to go back and report where all the changes being made are, release after release, and what the new strings are. And let me create a file and send that to the localization team, which then loads it up in their TMS, and let’s hope that the file formats are not “creative.” Let’s hope nobody’s missing a curly bracket, and let’s hope nobody forgot some set of files.

 And then, let’s hope that those are properly uploaded. Then, when they come back, let’s hope that nobody translated a file parameter that was accidentally listed as something that needed to be translated, or again, missed a curly bracket or broken encoding, or something like that. And then let’s manually put it into the build.

 And then let’s also consider that all of that manual processing might have involved somebody, like, getting to it on Tuesday rather than right away. So, we see clients that, you know, maybe had a localization turnaround time of like five weeks, which is really slow in this day and era. We see companies going from five weeks to like three days. That’s really common. Including all the translation work.

 Sometimes, quite a bit less than that. There’s both the i18n side and then just the simple file management. You’re talking about turning people into “file nannies” rather than high-value developers.

Renato Give me a sense of what is happening, because it sounds like 20 years ago, we were dealing with sorting orders and decimal separators and things like that. A lot of that is now in libraries. It sounds to me that what used to be externalizing strings and taking care of concatenations and things like that—which were the traditional, historic internationalization—it seems that these problems haven’t gone away, and it seems to me, listening to what you’re describing, that the process has become even more complex than it used to be.

 So, it’s one of those things where automation is not necessarily making things easier, it’s dealing with a higher level of complexity. So, when is the job of an internationalization engineer going to disappear?

Adam People don’t realize that just because a programming language technology or, say, a JavaScript library supports internationalization, it doesn’t come for free. There is no ‘automagic’ solution for this. I mean, you can have proxies and the like, but eventually there’ll be a millstone around any development team’s neck.

 With a JavaScript library like React or any of the other ones, there are ways to internationalize better. Unless the developer is actually taking advantage of them and knows the issue, you’re still going to run into a lot of trouble. You could say, “hey, why is security still a problem?” I compare it a lot to security because it’s a high-profile issue. You get a security problem and it can cost you a lot of money to fix and a lot of lost faith.

 Localization doesn’t tend to make the news, like you say, Renato, unless it’s really bad. But, the fact is, it’s a hit to people and it hasn’t gone away as an issue. Automation actually does help you scale, but it’s a different type of automation than just saying that Java supports Unicode naturally as a string. You know, that’s an extreme example, and that’s been around for a long time.

 Just being able to support globalization doesn’t necessarily mean that it works automatically. When I talk about automation, I talk about taking the human error and delay component out of something that happens over and over again, that is easily solved. In the same way, you might use translation memory—it’s a very basic technology, so you don’t have to translate certain words over and over again. You have errors that happen or tasks that happen over and over again that really can be automated. And those should be automated.

 Those are automated by many of our products from, you know, QA tasks to file management tasks, to simply performing good internationalization as you’re developing code.

 So, yeah, the traditional i18n issues, they haven’t gone away. And nor do I see them going away any time soon.

Renato They just change into different formats and different languages and different types of code, right? Because internationalization is a product design issue. It’s not a service or, like localization, that is something that happens after the fact.

Adam Right, right, it is. You know, to a large degree, when you internationalize, you’re changing your design framework, you’re modifying them, it can underpin your entire code base if you haven’t done it. It’s not an unknown R&D project; it has a known outcome, it has a known way to execute the outcome.

Renato I understand that you have integrated internationalization with localization with some tools that help you manage both processes. Is that something that you think is a direction that Lingoport is taking?

Adam It’s been our direction for a long time, actually. It’s really picked up steam in the past few years, as we’ve gotten both more customers and more mone y to, you know, internally push into our development project and a lot of feedback. So that’s really cool. The most important thing that I see is the demand on scalability. For instance, you can look at one SaaS-delivered product that has a web-based and two sets of mobile-delivered interfaces for iOS and Android and see that, you know, it’s not uncommon for that one product to have 150+ repositories that are being automated and managed so that localization is fast and really scalable.

 Some of our customers literally have tens of thousands of products, and when you have to scale localization across that industry, you just can’t afford to have project managers and localization managers for every one of those things messing with repeatable tasks like file management over that many possible changes. And so, you’re starting to get into this really massive scalability that’s happening within the industry.

 Most of the industry is still involved in lowering the cost of a word. Whereas, we’re saying, hey, there’s this whole world of software development that’s really unique, and that gap between development and localization, that doesn’t have to be there. We can make localization just another requirement. And it can even be automated to the point like, you know, when you walk into an office building, you kind of expect the lights to be on—well, because they come on automatically! That localization process can be just the same.

**End of conversation**