

EE Podcast #50 Documenting EE

Lea Alcantara: This is the ExpressionEngine Podcast Episode #50. Yes, you read that right or heard that right. That's 50 where we talk documentation with our guest, Erik Reagan. I'm your host, Lea Alcantara, and I'm joined by my co-host, Emily Lewis. This episode is sponsored by EECl 2011. EECl is up for its 5th season and this time it's returning to the United States of America the most significant conference where ExpressionEngine developers, designers and users will run from October 19th to the 21st at the Invincible Dog in Brooklyn, New York. A few tickets are still available, so check out EEClConf.com for more details.

Emily Lewis: The ExpressionEngine Podcast would also like to thank Pixel & Tonic for being our major sponsor of the year.

Lea Alcantara: So this episode is called Documenting EE, but really it's all about being a responsible developer, I think, in taking care of the contingencies. So, to help us talk more about documenting EE and how, we have Erik Reagan. Hi Erik, thanks for being with us.

Erik Reagan: Hey, thank you.

Emily Lewis: So Erik, can you tell us a little bit more about yourself and how you got started with EE, and what prompted you to start documenting your development steps?

Erik Reagan: Sure. I've been using ExpressionEngine personally for only a few years. What led to that was I was working at a church that was quite large. We had a number of websites out there and we had a need to support or manage multiple websites from a single control panel. So I started looking around for a solution that met that need, and I came across ExpressionEngine pretty early on that in that research and tried it out and absolutely hated it. It was driving me insane just simply because I was coming from a WordPress background.

Emily Lewis: Sure.

Erik Reagan: And before that, I had to use CodeIgniter to write some small custom CMS things, but other than that, I didn't have really any experience with any other content management system. So coming from that WordPress where everything is like a post or a page, coming from that background, I just didn't get ExpressionEngine, and I basically tinkered with it for a week or so and then just said, "Never mind. This is not cool."

Emily Lewis: Cool.

Erik Reagan: So about a few weeks later, I decided that, "Maybe there was something to this. The community clearly likes the product and it's being used quite a bit, so maybe I should spend a little bit more time trying to figure it out." And then I came across Mike Boyink's Train-ee series about building a business website in ExpressionEngine and also building a church website in ExpressionEngine. So I read through Mike's materials as I think many people have in the EE community, and then I kind of had that light bulb moment where I was like, "Oh, that's how it works. Okay."

And so I started to dive in as much as I could to learning about Ryan Irelan's Screencasts and I read as much as I could and read through the documentation and really started to really understand the product more and more, and I started to like it more and more. So we ended up using it for that site and those are multiple sites, and I'm very thankful that we did. At the same point in time, I was doing some freelance stuff and I began to use ExpressionEngine for all of my client's project and I basically haven't really looked back since then. At that point, I was trying to build a small CMS with CodeIgniter and realized that I had absolutely no need to do that because ExpressionEngine was what I needed and much more than I could have built on my own.

At this point, it was pretty close to the announcement of EE2 being on CodeIgniter and I was all excited about that. A couple of weeks later, I finally saw that coming to fruition. So that was kind of what initially drew me into ExpressionEngine was just the community that seem to rave about the product and then the fact that it was eventually going to be built on CodeIgniter, which I was already familiar with.

Lea Alcantara: So in regards to that, like when did you start being more serious about documenting your ExpressionEngine projects. Have you always been a prolific documenter in previous projects, even with WordPress?

Erik Reagan: Oh, by no means. I wouldn't even claim the word "prolific" yet.

Lea Alcantara: Okay.

Erik Reagan: But the kind of catalyst to this for me was when I actually left that position at that church. I mentioned I was freelancing at that time, and basically, I had a business name. I had a business partner who was a designer. I was the developer and we knew we wanted to have a business long term. We just weren't there yet, and so that was kind of goal all along was to get to the point where I could be self-employed full time, and once that day came, I essentially put in a resignation to the current job and I realized that I hadn't documented much of what I was doing and that the person who was going to be coming behind me was going to be just not in a very good place because of my lack of documentation for how things were set up, and that person might very well be listening to this at some point and nodding in agreement.

Lea, I've actually talked with him quite a bit about that and they've heard a number of apologies, but that kind of what spawned the initial problem was the idea that I'm not going to be the only person working on something, and even if I am, I might have to come back and look at something later and just not remember why I did something that I did. So that was kind of the initial catalyst.

Lea Alcantara: That's interesting what you mentioned about sometimes it's not even just for legacy, so if someone goes out and works on a project after you to make it less confusing, I forget a lot of the reasons why I did something.

Erik Reagan: Sure.

Lea Alcantara: And I've got a bit of a confession, sometimes I haven't been perfect with my documenting. In fact, it's next to non-existent. Mind you, I'm trying to become a lot better with it, but I don't know if it's necessarily a bad thing, but I'm mostly the main designer/developer for EE for most of my clients. So if I forget how to do something, but I know I did it before but I didn't document it, I just log into my old client's site and then just re-review that. But that's kind of bad because I'm trying to log into another site, even though I technically shouldn't be doing that because I'm not really doing anything. I'm just reviewing what I did when I should have documented that on my own to refer to for a new project.

Erik Reagan: Sure.

Lea Alcantara: Have any of you guys ever done that?

Erik Reagan: Oh yeah.

Emily Lewis: Well, I do that all the time, and in fact, most of the time. I mean, just from a simple copy and paste perspective, I'm like, "Oh, I did this on this previous project. Let me go into that source code and copy it and paste it." And not even referencing the documentation. It's a bad habit, a very bad habit.

Lea Alcantara: Yeah.

Emily Lewis: But I will say that regarding documentation, Erik mentioned handing over to someone else and also Lea was talking about revisiting, but I spent most of my career working on teams and documenting and even, I guess, a lesser level of documenting and just good commenting was a hallmark of how we did work on a team because files were constantly being passed between. If someone would go on vacation, then someone else have to pick up. So it wasn't even just when a project was finished and we needed to hand it over, but as a project was ongoing with different team members to know what the status of things were.

Erik Reagan: It's true.

Lea Alcantara: Did you find that too, Erik?

Erik Reagan: Yeah, absolutely. Similarly to you, Lea, for the majority of the time I've been doing development, I've kind of been the only person on a certain project and so the goal for me for documentation has more so been individual and personal. My goal is to eventually save myself time and money and to save my client time and money, but never really within the context of working with the team up until this year when that started to come into play for us and it's even more important at that point, I think, to have some kind of documentation within a project. It's where you can just bring someone new on who might not have done anything with that project, but needs to get up to speed quickly rather than telling them, "Okay, we are going to sit down for five hours and click through the site and talk about what does what, or okay, I need you to just go look at the code and try and figure it out." Rather than doing that and saying, "Okay, here is the documentation telling you where these certain key features are and how they work," kind of maybe at a high level and then sometimes even getting down into code samples and the logic behind why something was done.

So we are finding more and more about that team stuff now, and I mean, I'll be the first to tell you, it's an ongoing learning process for us, but we definitely wouldn't put documentation aside at all.

Lea Alcantara: That's interesting. So Emily mentioned commenting in code. Is there a difference between commenting in, like, while you code versus documenting, or is it really the same thing?

Erik Reagan: I think that they go hand in hand. Commenting in code can definitely be separate from documentation, because to me, there are kind of two different pieces to the puzzle. One of them is what I would just consider kind of a master documentation file or document, and then the other part of it is documenting within your files. So in the context of ExpressionEngine, that means documenting within your EE templates or maybe within custom add-ons you have to write and things like that, and then within the master documentation file, you would more include some of the structure of your content and your site.

I've kind of got this little list that I work from that I basically must include in my own documentation to make sure things are safe, and it's far more than just the site code. It even goes into server environments. If you are working in multiple environments, what does each environment support and things like that?

Emily Lewis: Yeah, I was looking at your slide deck from Engine Summit presentation that you did, and you were talking about the things that you documented starting with like the server environments. You also mentioned like your channels and related groups and even working with third-party add-ons. I'm just curious. Like when you start the process of documenting, does it actually start for you before development? For example, you've mentioned server environments. I mean, with all of that stuff, you are going to know in advance. So is it something that is part of sort of the kickoff of the project, the pre-planning and then it's sort of maintained throughout, or do you only start documenting as you are developing?

Erik Reagan: If I do it correctly, then it's always before. The goal of the documentation is to provide yourself a roadmap, and with any type of code, it's kind of weird area when you talk about building a website. If you are developing software, I think it's a little more of a clear picture because to me documentation really does need to come first with software. With a website, there is a little bit of grey, I think, just because of the tools that are out there to speed up a process and the types of processes that go into creating websites and the design aspect.

So documentation, for us, definitely needs to start before the development process and it just creates that roadmap. And whenever there is a custom add-on or something involved, it is absolutely necessary and even in more detail. But to touch on the other side, it's also maintained throughout the process of the development. So we will start with the kind of a core set of structure for the channels and custom fields and all that alongside maybe what add-ons do we intend on using. Before we build the site, we pretty much know the add-ons we are going to be using, so we will document those things.

And if there are any nonstandard, what I would consider nonstandard ExpressionEngine techniques that I'm going to use, I'll probably document that just so that if I have, like I said, somebody new on the team or if for any reason, we work with other contractors for a project, they might not build things the same way we do. So it's important to document some of the approach to what you are doing. I think we all kind of know that with ExpressionEngine, it's greatest strength is also its weakness in that you can do so many things in so many different ways and won't get the same result in the end.

Lea Alcantara: So I'm curious then with all the documentation that you are doing and trying to do, how have that affected timelines with your general development?

Emily Lewis: Oh, that's a great question.

Lea Alcantara: Because I mean, I think part of the reason why I guess it's laziness that I don't do documentation as well is because in my mind, they already have kind of track of the stuff that I wanted to do.

Erik Reagan: Sure.

Lea Alcantara: And then so I just go in and do it, but then I don't write it down or type it out, like how much more time has increased in order to accommodate documentation?

Erik Reagan: Sure. The beginning of a project for us is typically what we will just consider discovery with our clients and where it's mainly interviewing client needs and figuring out what is this site going to really entail or this project, and part of that now is documentation for us. So it's not always something that they see, but before we get too far into even to the design things, we will be working on the documentation. And it does bad time with the early part of our projects, but it will also save us time when it comes time to build the project out. So ultimately, it does add to the project, but it doesn't add a lot.

To give you a kind of an example, if we are building just maybe a simple brochure-type website with some small pieces of dynamic things for blog and articles and maybe some member commenting or something, we will probably spend about two days, and that's two full days, working on the documentation before we begin development.

Lea Alcantara: Wow.

Erik Reagan: And that's a small website.

Lea Alcantara: Interesting.

Erik Reagan: And so in the grand scheme of things, though, it's not a lot of time compared to what we spend on the entire project.

Lea Alcantara: Sure.

Emily Lewis: Now, this documentation...

Lea Alcantara: Do you part... well, sorry, sorry. Before we move forward with that, do you price that out for your clients? Do you add that as...

Erik Reagan: Absolutely.

Lea Alcantara: Yeah. Okay.

Erik Reagan: That's a feature. It's huge. So with clients, at least for the ones that we are working with, that is a very good thing that they hear. It's a strong selling point for us in that we have this extra level of responsibility and interest in their time and money, and we are very upfront in saying, "We would like to create a lasting relationship with you as a client. But in the long run if you need to work with someone else, we don't want to be hindering that process." We don't always want to support them in finding someone else to work with.

Lea Alcantara: Sure.

Erik Reagan: But we definitely want to be responsible and respectful of the time and money that they are going to put into something and supporting us with their business and choosing us as a vendor for their project or a partner on the project. So we want to make sure that we do things in their best interest and in ours, and that's an extra cost to them.

Lea Alcantara: Yeah.

Erik Reagan: But it's a huge comfort to them as well.

Lea Alcantara: Has any of your clients ever rejected that saying, "That's okay, I don't need that."

Erik Reagan: No.

Lea Alcantara: It's a selling point and they always pick it up, they always pay you.

Erik Reagan: Yeah, I've had nobody to say, "Oh, I don't really think you should spend that time on that." Nobody has said that to us yet.

Lea Alcantara: Okay. Cool.

Emily Lewis: This sort of reminds me of Leevi Graham's presentation at EECI last year in San Francisco where he talked about the pre-planning process that they do, and it's really, really in depth. And what he had talked about is not only was it a cost that the client was aware of and sort of part of, not only the fact finding and planning and all that other stuff, but it was also the documentation that they put together for their planning purposes at Newism is the actual deliverable. Erik, do you treat documentation in that fashion as a deliverable to the client? And I don't mean the client documentation of how to use the system, but the actual planning and the documentation of EE and the environment.

Erik Reagan: No, we have yet to even share that document with anyone other than our development team. We are happy to show it to our clients, but the format that we use is not necessarily going to be friendly to them. We use Textile basically for our documentation format and so we have a single file in the root of our website repositories because all of our sites are in version control, which is I think an important side note that we have a version controlled instance of our documentation. But we keep that in a developer-friendly place, but not in a client-friendly format. So while we are creating it, while they are aware of it, while we are happy to share it with them at any time, they rarely see it. I don't think I've shown any of my clients the actual documentation to this day.

Lea Alcantara: Interesting, because I'm looking into one of my major clients into doing giant documentation. I know after the fact, which is bad, but...

Erik Reagan: Better than never.

Lea Alcantara: Yes, exactly. But I'm thinking of doing it in a more public form, nothing like super complicated, but basically a blog with categories where we just type out whatever the documentation and whatever the subject is in associated to a category and have it publicly viewable. Now, I don't really necessarily think that my client cares. It's really just more for me and my team, but I find it interesting that while you say that you can make your file accessible to your client and it's not really a big deal if they ask for it, but your first choice is to make it not as easily accessible.

Erik Reagan: Yeah.

Lea Alcantara: Is there a particular reason why, or is it that what just happened in terms of workflow?

Erik Reagan: That's just what happens, specifically because they still receive other documentation from us in terms of the actual site management. And if they see evidence of that being produced, I mean in my experience, don't doubt that you've done the other documentation.

Lea Alcantara: Sure.

Erik Reagan: And so, with us telling them that we can give it to them if they need it seems to have been enough so far, so just supply some context. I really like talking about doing documentation stuff, but our company, we only build a few websites a year. So far we've only done a half dozen to a dozen or so of these, so there is not a large test base for us saying that nobody wants to see the documentation or anything like that.

Lea Alcantara: Okay.

Erik Reagan: But in our experience, it has worked that well in that they hear us in saying that it's there, but they haven't really expressed too much interest in seeing it. I guess maybe the level of scare we give them in saying, "This is for developers and you won't understand it."

Lea Alcantara: Interesting.

Emily Lewis: So Erik, I'm curious about the style of your documentation as well as content of it. So when I'm talking about style, I'm just curious, if you are using different flags or separators to delineate different types of content or an order in which the documentation starts at the top and continues through, so organization and visual presentation of the style. But then kind of how detailed do you get with your documentation? I mean, where do you draw the line, like how far down the rabbit hole do you go in terms of documentation?

Erik Reagan: Sure. Well, to start that off, the way that we separate out different sections of our documentation is just by leveraging some of the Textile syntax. So as an example, I use TextMate as my editor daily and it was a Textile bundle that does some highlighting and some formatting for me right within TextMate to show me kind of a visual hierarchy of things. I can, with TextMate, also preview that Textile document as rendered HTML or I can even convert it to HTML, which sometimes helps read through some of the longer document files. But in addition to that, it is actually some

interesting timing. About a week ago on a Friday, I had this idea to basically make a small add-on that took our documentation flat file and parse it into an ExpressionEngine control panel-friendly format.

Emily Lewis: Oh.

Erik Reagan: The reason that came up, I guess, this is kind of a side note, I've talked to a number of people about the way that we do our documentation. Some people really liked the idea that it's a flat file and tracked in version control and developers can just open it right there within the repository. But other people are like, "Well, I want it in the control panel. Why don't you just make it into a separate channel with ExpressionEngine and build out some templates to show you documentation that way."

And really my stand on that has been that the site might go down sometime. What if you need your documentation because your site is broken, and then your documentation is in the broken site. So I never wanted to lean on that.

Lea Alcantara: Good point. I never really thought about that.

Erik Reagan: So in the past, I've just essentially sat on this flat file, but for whatever reason, about a week ago, I was thinking through the idea of making this flat file accessible within the control panel, so we could have the best of both worlds. So now, I sat down for a few hours and put together a little add-on, actually named DevDocs, that takes my Textile formatted documentation and spits it out into a control panel-friendly HTML almost paginated type documentation. So it basically just looks through the Textile document for each one tag and makes that a separate page.

Emily Lewis: Oh, wow.

Erik Reagan: And the reason I did that is specific to a project we are now where it's a pretty involved project. So the documentation was getting pretty long and it was also a team project where I needed some folks on board who I didn't initially know I was going to need in our planning process. So I needed a way where I didn't have to run through, "Okay, this is where our documentation file is and this is how we are going to be working together." Instead I was able to get this little add-on together so that everybody now, instead of referencing the flat file, I think everybody go to source this just to the control panel, and I'm the only one probably using the flat file now because the control panel is actually pretty.

Emily Lewis: Yes.

Erik Reagan: It's visually stimulating and sometimes the text document is, even to other developers, sometimes they want to get out of the text editor.

Emily Lewis: Are you going to be making that add-on available to other people or doing blog post talking about it at all?

Erik Reagan: It's already up on our GitHub account. So if you are curious at taking a look at it, it's [GitHub.com/focuslabllc](https://github.com/focuslabllc) and it's just called DevDocs. And I threw it up just so that people could take a look at it. It's going to be up there forever. I mean, we are not trying to sell this or anything. It's just part of our development process. With tools for our development process, we are not necessarily in a situation where we want to sell something with that. So it's up there now. It only supports Textile formatted documents, but we are going to actually make it a little bit more robust where you can mark

down or a few other approaches and make it a little more customizable so people can use it without our specific format requirements. But for now, that's what it does. It's pretty basic and it's working really well. It also caches your documents, so you are not always trying to parse a file on that page load, but that's a little probably more technical information than we wanted today.

Lea Alcantara: I have a question in terms of...

Erik Reagan: Sure.

Lea Alcantara: You mentioned that the flat file was version controlled and et cetera and so forth, but all your documentation is in one flat file. So then how do your team members understand which part of the documentation has changed?

Erik Reagan: Well, that's just the beauty of version control. And that will vary based on someone's experience with using version control. You can always run a diff or difference against one revision to another and it will just highlight the things that are changed. So being familiar with the version control process makes that pretty simple.

Lea Alcantara: So in terms of the control panel add-on that you created, is that reflected when you are changing or not at all then?

Erik Reagan: That is not reflected in the control panel unless I intentionally do something. So for example with Textile, I can add a class to a line of text if I want to by wrapping a word in parenthesis and putting it in a certain spot and then it will give something a class. So we have a class of NOTICE and it's basically just my way of flagging it for somebody right now. It's kind of experimental way just because it's only something we've been using on this most recent project since we are finally in the control panel with the documentation rather than just a flat file. So I'll do that. If something is new and needs to be done or reviewed prior to that notice going away. So it's working so far, I think, and it's maybe not the best approach, but it's not something that we've had a need for it until now. So we are kind of experimenting with that same changes from the control panel side of that documentation.

Emily Lewis: Yeah.

Lea Alcantara: So I guess that's a bit of a benefit in the other approach where let's say you just create a blog or a new channel with the entries. Because if you have one entry per like little thing that you are documenting, if you change one thing, it's very clear in the control panel just in EE or whatever system you are using if you wanted to show most recently updated entries, that shows up right at the front without you having to go into flat files or anything because that's how the system is built. Would you say that's true?

Erik Reagan: Yeah, I'd say that's true.

Lea Alcantara: Yeah.

Erik Reagan: And if that's the desired workflow for you, then a Wiki-style documentation is probably best. I favored the flat file approach, but in the end, it doesn't really matter how you are documenting your project as long as you are documenting it.

Lea Alcantara: Yeah.

Erik Reagan: So if you have more of a Wiki workflow or a channel entries workflow, then more power to you. It's definitely still works. So it's just a matter of what works for you, what works for your team and running with that.

Emily Lewis: And as you are figuring out, how do you know how much documentation to do? Like how much is too much? I mean, is there such a thing as too much and it becomes an ineffective use of time?

Erik Reagan: I think that there is such a thing as too much, but I don't think I've met that point yet, so I'm not too sure when that hits. It's hard to say just because of the range in projects that we work on. Some of them are just ridiculously simple and the only reason we use EE is because we know it better than WordPress. It's one of those things where the documentation is very, very light in that they have a blog and they've got a home page with some stuff that's customized, but really they don't use many channels. We still document those and it doesn't take much time at all, and usually those are smaller documents. And then we've got sites where we work with numerous outside vendors and numerous custom add-ons and have four or five environments and 15 different channels with content and 20 channels sometimes. That's when it gets to the point where you want to consider am I documenting too much, but we haven't really hit that issue. So I'm not really sure what that would look like or feel like. But I do think that that point does exist.

Emily Lewis: And do you happen to have an opinion? I was doing a little bit of research before we talked to you. I found a lot of stuff online. There happened to be some people out there who, I don't want to say their anti-documentation, but they are more pro-self-describing or self-documenting code as opposed to you creating a separate document for your documentation, but writing your code, your markup, your CSS or whatever it's going to be in a very self-describing way as being the preferred way of documentation. Do you happen to have an opinion about that?

Erik Reagan: I haven't seen or heard it, but I have an initial reaction to it. I don't like it in the context of ExpressionEngine.

Emily Lewis: Sure.

Erik Reagan: Again, it just goes back to how many ways we can do one thing in ExpressionEngine, and ultimately, I might bring somebody on my team who has never used EE, but they have a really strong PHP and MySQL background and I need them to kind of understand some of the pieces of ExpressionEngine without maybe sitting down with him and clicking through everything. In that case, they might not know where to look for these templates that might be self-documented or whatever. So I still see a lot of value in having a single outside file that's kind of a overview of the project at large.

Emily Lewis: Yeah, my initial impression of that sort of idea of self-describing is that it seems to me it's probably a good practice to support good documentation.

Erik Reagan: Sure.

Emily Lewis: Where you can be self-describing, even with something as simple as your class and ID naming for HTML or CSS, be self-describing so that it's clear to someone or yourself later. But yeah, my personal feeling is that it probably can go hand in hand with good documentation, but I can't see a way of making it or replace it.

Erik Reagan: Yeah, I think that this self-describing is a good practice in general, even if you are not doing documentation. I hadn't really sort of heard that phrase, but it's definitely a coding style that we encourage and require from people we work with. After I gave that initial presentation at the Engine Summit, I came across an article by Tom Preston-Werner, who is one of the co-founders of GitHub and he has a stance on, essentially, I don't know if he coined the phrase or where it came from, but it's readme driven development, so there are a lot of development processes that are abbreviated with something, fill in the blank driven development. So if you are talking to Steven Lewis, you will probably hear him talk about test-driven development. There is also behavior-driven development and a lot of other ones, and so there is this idea of readme driven development, which is you don't start your project until you've written a readme file.

Lea Alcantara: Interesting.

Erik Reagan: And so I found that, and I found it to be very much in line with what I was trying to think about and he actually discussed that there are some differences between the idea of readme driven development and documentation driven development, whereas documentation driven development is more of you create this huge file that covers every little detail and then start, which is kind of a waterfall programming approach and then compared that to what he says is readme driven development, which is much more high level, cover your bases, talk about what you are going to do and then do it rather than every little detail.

But I think that even within any of these development processes, you have this kind of self-described approach of writing code, which is just a good practice in general. So if any of you are working on websites or especially add-ons or anything, there is a really good resource on the Pragmatic Bookshelf and it's called the *Pragmatic Programmer*. It's kind of like the little bible that teaches some programmers. That's something that I think maybe will be beneficial to a lot of the listeners to check out.

Lea Alcantara: So I think we are going to wrap up. Before we go, I do have one last question. Do you have any tips for our listeners over what particular parts in ExpressionEngine you think they have to, like no questions asked, they have to document these parts?

Erik Reagan: Templates.

Lea Alcantara: Templates.

Erik Reagan: If you start anywhere, start with documenting your templates from the inside out, and I mean, the beauty of ExpressionEngine is that you've got these comments that you can add in EE commenting versus HTML commenting in that it's not ever displayed in the HTML that's produced. And writing comments to yourself and other developers directly within the template is really, really beneficial. So my official stance on the template stuff is that if you only take away one thing from the documentation approach from me is that leaving comments in your ExpressionEngine template is a must, regardless of your level of experience just because of the benefits you get from there. Some people don't like the idea just because it messes with the white space in the rendered HTML.

Lea Alcantara: Sure.

Erik Reagan: But my stance is that that doesn't matter at all.

Lea Alcantara: Cool.

Erik Reagan: So yeah.

Lea Alcantara: Emily, do you have any final questions?

Emily Lewis: Nope, that covered it for me.

Lea Alcantara: Okay, perfect. So thank you so much, Erik, for joining us.

Erik Reagan: Sure. Thanks for having me.

Emily Lewis: Yeah, thanks. We really appreciate it. Before we go, if our listeners wanted to follow up with you, where they can find you online?

Erik Reagan: I'm on Twitter @erikreagan. Company-wise, we are at FocusLab, LLC, and that's pretty much where I am lately. I'm not on Twitter much because I've been a little busy, but if you want to catch us online, we are at FocusLabLLC.com.

Emily Lewis: Great.

Lea Alcantara: Perfect. So I'd like to remind people that we do regularly post topics and schedules on ee-podcast.com/schedule. Be sure to check it out. On our next episode, we will be talking to various nomadic EE designers and devs and how they fit in their workflow with their travels.

Emily Lewis: And in the meantime, if you have any ideas, resources or questions you would like us to discuss during the upcoming podcast with the EE gypsies, email us at feedback@ee-podcast.com.

Lea Alcantara: Now, we would like to thank our sponsors for this podcast, EECI 2011 and Pixel & Tonic.

Emily Lewis: We would also like to thank our partners, EllisLab, EngineHosting and Devot:ee.

Lea Alcantara: And thanks to our listeners for tuning in. If you want to know more about the podcast, make sure you follow us on Twitter @eepodcast or visit our website at ee-podcast.com. This is Lea Alcantara.

Emily Lewis: And Emily Lewis.

Lea Alcantara: Signing off with the ExpressionEngine Podcast. See you next time.

Emily Lewis: Cheers.