Jackie D'Elia: Hey everybody, it's Jackie D'Elia with another episode of Rethink.fm for you. Today, I have my guest, Matt [Pritchett 00:00:09]. Hey Matt, how are you?

Matt Pritchett: I'm doing well, Jackie, how about yourself?

Jackie D'Elia: I'm well. Thank you very much for joining me. For those folks who don't know who you are in the community, would you introduce yourself and tell us what you do?

Matt Pritchett: Sure, my name is Matt Pritchett and I hail from Atlanta, Georgia. I am a UX developer at Lift UX. We are a small agency that focuses mainly on UX development and design. We're scattered all over. We're a remote first workplace. We're scattered all over. We've got people in Florida, and Texas, and Michigan. I'm the only person in Georgia for now, but we're scattered all over the place.

Jackie D'Elia: Very cool. When did you start as a developer?

Matt Pritchett: I took my first developer position in August of 2007, straight out of high school, actually. I worked for a small missions non-profit that sent high school and college students overseas. I took the position because I was interested in non-profits and the church world. I spoke fluent Spanish at the time and I know I don't speak it fluently anymore. At the time I did and I served as a translator for them and took teams overseas. Part of the role and how they were able to hire me was I was the director of multimedia, which basically meant I took care of hardware and software. I had to learn how to do websites. I started out with table based development in Dreamweaver, like a lot of people did, and taught myself from there.

It's been a journey ever since.

Jackie D'Elia: You didn't start off developing in WordPress then. When did WordPress come into the picture, and how big of a part of it is in your daily work now?

Matt Pritchett: In August, I started that position. By December of that year, part of what my role was they had this ancient PHP system that allowed teams, when they were overseas, to upload these text and picture based updates so parents could keep an eye on their kids while they're overseas and make sure they're not in harm's way or anything like that, but this thing was so ancient that anytime you touched it, breathed on it, looked at it the wrong way, it fell apart, it errored out, it deleted things. Honestly, I found WordPress because I was in trouble with my boss for having deleted a bunch of the updates from previous trips on this system. I was like, "I don't understand how this system works. We need to replace it." I'm in trouble, I've got to figure out a quick win, and I came across WordPress and it was the answer to a prayer, almost.

It was easy to use, we could do exactly what we needed to do, it worked on terrible third world internet connections. You could use it on slow internet connections. Started doing, basically, blogs from there and we built several custom themes and that led to building custom websites on WordPress. From there, I took a couple of freelance positions. I moved on from that job and went back to college and put myself through college waiting tables, but also freelancing on the side. From there, I actually took my first development position out of college as a WordPress developer working at a very large church. I was the only developer on staff so I got to do things however I wanted to do. I learned a lot of bad habits that later were corrected.

From there, I actually took an agency position as a WordPress developer, and since then I've taken several more agency positions, but they've all been in WordPress. WordPress is 100% of my job these days. At Lift, we're actually doing some pretty cool things with WordPress that aren't just typical HTML, CSS, JavaScript, PHP. We're doing some different things with more JavaScript frameworks and those kind of things, but it's still mostly WordPress.

Jackie D'Elia: That's a lot.

Matt Pritchett: Yeah.

Jackie D'Elia: It sounds like your experience with WordPress is very similar to a lot of other people that just started off working in it and then tried to formulate a workflow and improve your work habits as you go working with it. I've had the same experience. We wanted to talk about local development environments today. That was the inspiration for us chatting. We're in a mastermind group together. I know you've been rethinking some things. I thought this would be a great opportunity to just talk about it. There's many different ways to do things, and I think, for some folks that are just starting off, it can be daunting to figure out what direction to go and how to do things. I know I wrote a post a couple months ago about my code journey for the past three years. When I first started on that journey, I wasn't developing locally. I was talking about how I transitioned into developing locally and why, and what impact that had on my workflow and everything.

Basically, the reason we wanted to have this chat was to just talk about that and just talk about where you see the future of local development environments going and what role you want to play in that.

Matt Pritchett: When I started, I definitely did not start out with a local environment. It was cowboy coding all the way. Obviously, that's not something I recommend now. I think a lot of us, that's how we got started. It's not something I recommend today, but at the same time, everyone gets into this a little bit differently. I think I started working with local environments probably some time in 2008 with xampp at the time. Back in those days, you had wamp,mamp, and xampp. Those were pretty much your only options. You didn't have things like Vagrant, or ServerPress, or Docker, or any of those kind of things.

I got started with that. It worked, but it was fairly clunky. A lot of the times you had conflicting port problems. There were just a lot of issues with it. It got better over time. New technologies came out. I remember the first time that I installed Vagrant. It was amazing because you could configure everything. It took forever to get set up on it. I remember trying to convince our department at the time to switch over to Vagrant from a homebrewed installation of PHP and MySQL that we could basically pick up environments and put them on everyone's computer and they would be the exact same and how awesome of a concept that was, but it was such a hard sell because I, essentially, had to take every developer into a conference room, one at a time, and get them set up on this Vagrant box, because there were so many bugs in it at the time. It would take an hour, basically, for every development.

At an agency, an hour of a developer's time is very expensive, especially when you're talking two developers and my time, basically, taking a department of developers to do it, it was way too expensive. Vagrant's gotten much better since then. You have projects like VVV that WebDevStudios started, and that's being improved by the community now. You have projects like Mercury by WP Engine, which is a great Vagrant box and mirrors their own environments very closely.

Jackie D'Elia: On that point, can we just roll back for one second?

Matt Pritchett: Sure.

Jackie D'Elia: For some people, like when I just started on my journey, I'm using, right now, for my local development environment, I'm using Desktop Server. In this install that I've got, I'm fixed on a PHP version. I can't change my PHP version for that environment I'm running. In this way, yeah, I'm similar to what typical hosts I'm using, but I don't have that flexibility to customize my local environment to match an environment that my product is ultimately going to be deployed on. Some of the projects that you just mentioned, it sounds like there's two different ways that this is handled. Some where you can actually configure your box, your local development environment, to emulate where you're going to be hosting your site later, and that's an advantage. Can you just talk about that?

Matt Pritchett: Yes. Essentially, at the current time, you have two silos of local development environments. You have ones that are very configurable but are mostly command line based, so you use something like Terminal on Mac to start them and to configure them. You have to edit [.com 00:09:55] files and all this kind of stuff. It's very back end developer-y system admin focused. Then you have the class that essentially you have a nice user interface, a GUI, something like that, something like a Desktop Server, and those are, for the most part, and I think that's something that's changing, but for the most part currently they are very static. You don't get to change PHP versions on the fly. You can't select between Engine X and Apache.

Doing things like remote tunneling is most likely outside of what they do, and editing them is either impossible or very difficult. You really, right now, you're stuck between you're either responsible for everything and it's very configurable, but it's very difficult to configure for, say, somebody just getting started or someone without a lot of command line knowledge, or you can have something that's very easy to use, but not very configurable, especially on the fly. We're started to see a migration away from that with things like Docker.

Jackie D'Elia: What's that? I'm not familiar with it.

Matt Pritchett: Docker is a newer technology. It's been around for, I think, a couple of years now, but it's really just starting to go mainstream. Docker is a series of tools and an application, I guess you could call it. It can be a little hard to define. It's the idea that everything in your environment should run in a container. It should have its own box. PHP should run in a box. MySQL should run in a box. Apache should run in a box. You can take these boxes and package them up into an image and then just run that image, and run it anywhere. It's very configurable. It's very interchangeable. You can change out boxes on the fly. It's a way to compartmentalize things and then be able to change and add and adjust those boxes as necessary, quickly, and be able to reproduce those things quickly.

Docker just released Docker for Mac which is probably the simplest way to setup Docker for most Mac users, which I think most of the WordPress community are Mac users, but they do have for Windows and for Linux. Docker for Mac, you install the application and literally you run one command and you're ready to go. Docker has a huge library of different environments that you can run. They have a WordPress one that you download it and you have a WordPress environment that's ready to go, that's completely Dockerized, as you say.

The nice thing about that is it's taking all the benefits of Vagrant, with the configurability, and you can edit things, their .conf files. You can get access on the command line, but it's packaged it in such a way that anybody can work with it. You're able to switch out PHP versions on the fly, you're able to scale it up to thousands of machines if you need to, but at the same time, you can do it all from a user interface that's really nice and easy to use. It's still got some challenges to overcome. The community itself is still relatively young. There's some interesting issues on the licensing. There's a for-profit company that owns Docker, but it is technically open source at the moment. That's still playing itself out in what that looks like, but it is a very useful technology that I'm looking forward to what it has in store.

Jackie D'Elia: Based on that, what would you describe your ideal local development environment to be for somebody who's freelancing and then versus somebody who's working on a team in an agency environment?

Matt Pritchett: Those are very different. That's an excellent question. Your environment and the way you develop affects how you setup that environment. If you're working on a team, you need to be able to replicate code between team members and have very similar, if not exact the same environments so you can duplicate bugs, so you can all see the same thing, and to do so based on the environment you're developing for. You still need to do that as a freelancing, but I would say it's less important. You can do things however you want, whereas, obviously, on a team you have to come to some sort of consensus.

As a freelancer, which I do freelance work, I've used a little bit of everything, however I would probably say my favorite currently would be something easy to use, a Desktop Server, a Pressmatic, something like that. Something that's easy to use and it doesn't require a lot of my time. Whereas, on a agency level, I would say something that is easily configurable because I tend to do a lot more enterprise-level clients and their production environments are a little bit more complicated and we want to be able to match those because we see a lot more environment level bugs than you do on small non-profit side, a small business site. When you're talking about things like load balance or [caching 00:16:06] and those kind of things, you want to be able to mimic those environments as much as possible. That's when I suggest pulling in either a Vagrant or a Docker or something like that.

Jackie D'Elia: That makes it much easier in the agency environment to develop in the environment that it's going to end up running on, so it just makes it easier to do better unit testing and making sure that everything matches and works properly. In a freelancing local environment, you're typically on some Apache server, like a Siteground or WP Engine. There's not that much differences in those environments, I guess. Using something like a Desktop Server, or Pressmatic, or Mamp Pro, or one of those products, sounds like that's going to be okay.

Matt Pritchett: I think there's a little bit of ... You want to save as much money as possible, as a freelancer, because your time is money. As an agency, especially in the enterprise, you're actually going to save more money getting the environment as close to production as possible on the front end, to save time in the back end. Whereas, I don't think that's always going to be the case as a freelancer.

Jackie D'Elia: Next question I had was, just talking about one of the next challenges I'm going to be having in my workflow is pushing my local changes to a staging server, or to a production server. I'm building out locally. With Desktop Server, you can push the whole site, but then you have to push everything. Every time you make a change you've got to do this push. You do have an option to just push the database only, and that's fine, and then you can go in and push your file separately through SFTP or SSH and go in and push those up, but ultimately it would be great for the next step would be to be able to push your changes through some task runner or something that would just automatically push those up to whatever destination you want them to go to.

Where's things going with that? What options would I have, as a freelancer, to do that?

Matt Pritchett: Sure. There are lot of options out there. I'll say right off the bat that I think there's a lot of room for growth in this area, which is the reason we see projects like VersionPress. VersionPress is a new up and coming project. It's actually been open sourced quite recently. They're working on essentially GET versioning for database improvements. Every time you would write an article, you would essentially check out a branch of the database, and you would write it and make changes and make, essentially, commits, before merging that back in. It's something that I would not use in production quite yet, but it is making some really strong headway on something I'm definitely keeping an eye on.

However, tools that are really available now and that I use everyday, you've got a lot of GIT based deployment things that you can use. I know WP Engine has a lot of great tools that are built into their product for both migrating code and migrating the database. They've got a plugin that'll actually migrate your site for you. That can be used. I use Migrate DB Pro a lot, by Delicious Brains. Those guys are awesome.

Jackie D'Elia: I love that. I love the media files, being able to sync up those media files and push those up. That solves a big chunk of my deployment. What I'm only left with, then, is my code. If I'm building out a child theme, I've got my child theme folder that I need to pack up for distribution and then push that up. That's my point right now where I'm looking for how can I automate that process?

Matt Pritchett: We use a thing called DeployBot a lot, which is a GIT based series of commands. You can set it up several different ways. One is to run a git-push to a certain branch, and that branch is then linked to an environment. So you can have a staging branch that pushes to staging, you can have a QA branch that pushes to QA, a master that pushes to production. That's certainly a way to do it. We use that quite often and there are several services out there that do that. GIT has its own deploy system that you can work with. I'm not as familiar with that one. We don't use it just because my initial research indicated it wasn't quite ready for prime time. A lot of manage hosting companies like WP Engine, Pagely, and these kind of services have their own Git hooks already ready to go.

They have instruction tutorials and lots of documentation how to integrate with those. It's just a git-push to a certain remote and it automatically deploys and runs all the scripts and, a lot of times, it'll even validate your PHP and those kind of things for you.

Jackie D'Elia: You're pushing your code up to your remote repository, and then that's triggering something that would then deploy that code, those changes, to your production or staging server?

Matt Pritchett: I'm sorry. On WP Engine, you would push it to your own repo, say on GitHub or something like that, but then you would have a second remote. Instead of origin, you would setup staging or production, and then you would tie that remote to WP Engine's URL's that they provide you. You're pushing to origin and your server separately, as opposed to something like DeployBot where you push and then it hooks off of GitHub to deploy that. It's separate processes.

Jackie D'Elia: Okay, that looks like that would help solve that challenge of automating that process so that you're not ... My biggest pain point, I guess, in that process is I sometimes forget to upload all the files that I need to. I've got to go through each folder and make sure ... If I have a includes folder or a JS folder and I've made some changes there, I've got to remember, I've got to look on the dates and make sure I push everything, drag everything over in my FTP program to sync that up. This looks like that would solve a lot of those issues. The other question that I really wanted to talk with you about is you've mentioned that you're wanting to build a product that solves some of these challenges for a local development environment. Can you talk about what you're working on and what's happening with that?

Matt Pritchett: I can. You're getting a world exclusive here, I guess. I'm working on a new product. We've been talking a lot about local environments and deploying code and different hosts. I saw an opportunity to meet all of those things in the middle. I am going to be releasing, very soon, a product called Anchor WP. Anchor is a local WordPress environment, but with a lot of features that I think are going to be very useful to the community. Not only does it provide you a user interface that allows you to build Dockerized environments so they're very fast, they don't take an hour to setup, it doesn't take 40 minutes every morning to raise the environment. They're very fast, but they're also very modular. You can switch out PHP versions on the fly. PHP 7 is something that's up and coming, and a lot of WordPress developers really want to switch to it, but they don't have the ability to do that locally, or in their production environments. A lot of hosts are starting to integrate it, but a lot of local environments are falling behind.

If you want to switch between 5.2 and 5.6 and 7.0, that is something we can do because of Docker. You can switch between Engine X and Apache on the fly. You can basically change anything about the environment that you want to on the fly, and all it takes is a simple restart of the server. One click and you've changed the variables of that environment, all through a nice user interface. What I'm really looking to do, and what the real motive behind the product is not to just create another local environment product, but what I really want to do is be able to integrate with hosts and be able to integrate with themes and plug ins. You don't have to go out to get those things. You can mimic the environments that you're looking to build for, your WP Engines, your Pagelys, your SiteGrounds, your Bluehost. You can mimic their environments, even as a freelancer. You don't have to spend a bunch of time trying to figure that out, it's just ready to go.

You're not having to figure out these weird server environment issues later on, down the road.

Jackie D'Elia: That sounds like a really awesome thing to have, to not have to do all that research. "Okay, I have to develop for a site that's going to be on WP Engine" or "a site that's going to be on Pagely, what's their environment like?" Trying to figure all of those variables out. Like you said, some of the other products, you've got to be like a sys admin to go in there and configure everything to get it to work properly. For somebody like me that, maybe, is not going to be wanting to devote that much time for those things, that sounds like that would solve a really big pain point for most people, is to be able to say, "I'm developing on Siteground" or "developing on WP Engine and here's my ... " you just spin up the environment.

Matt Pritchett: Essentially, what I'm looking to do is to make a product that works for all levels of developers. We want to be able to take people who are just getting started and teaching them why local development is so important, but also make it easy enough to where they can actually get it and understand it and use it. At the same time, we want to make it configurable enough to where people who are working in the enterprise and need granular control over everything, PHP memory limits, and all that kind of stuff, to where it's configurable enough where they can do that.

If they know how to set their own environments up, they can templatize those things and have them ready to go and have them where they can just plug them in. We're also looking to partner with several plug in and theme developers, to where you can have templated environments with certain themes and certain plug ins, and it's one click install, and then you can templatize that and have it ready to go for a future project.

On top of that, one of the things we're looking to do is not only mimic hosts, but also be able to integrate with their deployment system. If I know that I'm going to be using WP Engine, Anchor knows that as well, as has a one click deploy up to WP Engine. That's something that we're still working on and it probably won't be version 1, but it is something that we're pursuing, and it's very important. I want to be able to create a product that's just not one little piece of the puzzle. I want it to be able take you from idea all the way up to pushing to production and launching the site, encompassing as much as we can in order to smooth out the process as much as possible.

Jackie D'Elia: I think it's very smart to integrate and to partner with others to bring in their pieces, versus trying to build it all out yourself. As new players emerge, and as new things change, there's more opportunities to just partner and put up the configuration and be able to just choose that one that you want, versus you having to build that all out and maintain it yourself.

Matt Pritchett: Exactly. One of my favorite things is I see guys like [Cory Miller 00:28:35] and [Chris Slumma 00:28:36], that repeatedly say this, "That I'm not the smartest guy in the room, but I do know those people." That's how I feel. I'm not the smartest guy. I know that. I always want to be the first one to say that. I know that there are people out there smarter than me, that have better products than me, and I want to bring those people in and say, "Let's work together. Let's integrate your things so we can all profit from it." I very much see Anchor WP as a platform in order to integrate other products and partner with those products so we can all benefit from it. Yeah, I am very excited about the possibilities of the partnerships that we're looking into, not only for myself but for everyone, for the community at large. I'm very excited to see Anchor get launched and get feedback, and hopefully help a lot of people.

Jackie D'Elia: I'm looking forward to being one of the first beta testers, for sure.

Matt Pritchett: We'll get you signed up.

Jackie D'Elia: All right. Okay, well we've only got a couple minutes left, but I wanted to ask you, any favorite local development tools? I know you mentioned you're using Migrate DB Pro. I use that too. I find it very helpful for my workflow. Anything else that you can share that's what's your mix of tools are right now?

Matt Pritchett: Highly recommend a task runner, whether that's Gulp or Grunt or Webpack or Webpack with Gulp and Grunt. Webpack is a new one for us that we're exploring a lot right now. It's looking super awesome. I can't overstate how much a task runner has saved me on time where I don't have to compile things or [lint 00:30:23] style sheets or beautify them or put them in correct orders or anything like that. I also highly recommend PHP Codesniffer, it's essentially a [linter 00:30:33] for your PHP, it makes sure that you're following the WordPress core standards, which is a huge thing and will keep you out of a lot of trouble. Basically anything that will automate my workflow is a thing that I'm looking at using. The more I don't have to think about, the more I can think about writing code that actually works and makes sense and does what it needs to do and is bug free.

Jackie D'Elia: Are you using Atom as your code editor right now?

Matt Pritchett: I do. I've pretty much tried them all over the years and I've settled with Atom for now. I always want to be flexible in that regard because being dogmatic about a code editor to me doesn't make a lot of sense because they all have pros and cons, but right now, Atom is the best one for me for the time being.

Jackie D'Elia: I can definitely recommend the PHP Codesniffer. I have that setup in Atom and in PHPStorm, that I've tried both of those. It is really nice to have that [linting 00:31:35]. I think I've set mine up to, whenever I save the file, it goes ahead and lints it. You can actually set it up to lint on the fly as you're typing, too.

Matt Pritchett: You can.

Jackie D'Elia: But it is nice to have it format everything properly, make sure that you're following all the standards, and it saves you a lot of time. I know you can even do it from the command line, too, so you can lint your entire folder with all your PHP files in there at once, and it gives you a printout of where you need to make all these corrections and what you need to do. I would recommend that to anybody that's not using that, that could save you a lot of time, and also make your code more presentable.

Matt Pritchett: Totally agree with that. I think the number one way to improve your code and improve your chops is to get it reviewed. I think the first place you should get it reviewed is by the Codesniffer. It will improve your coding overnight. It yells at you pretty hardcore.

Jackie D'Elia: Okay, so for my last question, if somebody isn't working with a local development environment yet, and wants to start, what would be your recommendation?

Matt Pritchett: I would say the easiest and probably cheapest way to get started right away is to go with something like Mamp, the basic version. I think Mamp Pro is awesome, but one of the best things about Mamp is that it starts out free. It is super easy to setup. It's a super easy interface. I think, very soon, the easiest way will be Anchor, but I'm a little biased. Probably, as of today, the easiest way is to use Mamp. It's been around for forever. I can say that tens of thousands, if not hundreds of thousands of developers have gotten their start with Mamp. It's still a great option.

Jackie D'Elia: That'll wrap up this episode of Rethink.fm. Matt, I want to thank you very much for joining me. If folks want to follow you and learn more about your upcoming product, how can they reach you?

Matt Pritchett: Yeah, so you can follow me on Twitter. I'm @MRPritchett. It's P-R-I-T-C-H-E-T-T. Or you can find me at my blog at blog.pritchett.media.

Jackie D'Elia: Thanks again, Matt.

Matt Pritchett: Well thank you, Jackie.

Jackie D'Elia: Everybody have a great week and I'll see you on the next episode.