EE Podcast #61 Version Control with EE

[Music]

Lea Alcantara: This is the ExpressionEngine Podcast Episode #61, Version Control with special guest, Adam Wiggall and Ian Pitts. I'm your host, Lea Alcantara, and I'm joined by my co-host, Emily Lewis. This episode is sponsored by mithra62, Backup Pro developer from mithra62 allows you to install Backup Pro on an unlimited number of ExpressionEngine sites to provide comprehensive disaster recovery for all of your clients and projects for less than the cost of a single ExpressionEngine license. With Backup Pro, you can be confident you have your clients protected and covered when disaster strikes. Available and supported exclusively through Devot-ee.com.

Emily Lewis: The ExpressionEngine Podcast would also like to thank Pixel & Tonic for being our major sponsor of the year. [Music ends] So Lea, I am really, really excited about our topic today. We are going to be discussing version control. But we have special guest, Ian Pitts and Adam Wiggall. Welcome guys.

Ian Pitts: Hi there.

Adam Wiggall: Hi there, thanks for having us.

Emily Lewis: We are really excited. I know both of you all personally and I've worked with you both, and in fact, it came up as a result of a conversation I had with Ian over chat about what Ian and Adam were doing with version control for their workflows, and I thought it would make a fantastic topic for the podcast.

Lea Alcantara: Absolutely. I mean, I think version control is something that the community has been talking about for at least a couple of years now, and still though that it's kind of new for the majority of the community, I would say. I think the reason why I would say that is that ExpressionEngine designers and developers, it is s a vast community, but a lot of us come from a front-end design background. So stuff like version control is something that's very new to us, so that's part of the reason why I'm really excited about this topic and hopefully we get to know a little bit more about the process on how we can make our workflow a lot more secure and easy to update our sites.

Emily Lewis: Yeah, and I've not actually done anything with version control. I've done some reading and I've attended a couple of presentations. So I'm just curious, Ian, if you could just briefly give like an elevator pitch explanation of what version control is, at least, with regard to how you guys are using it.

Ian Pitts: Yeah, actually, it's kind of a funny aside. Adam is the one who turned me on to version control, and I was actually pretty resistant at first. I didn't know anything about it.

Lea Alcantara: [Agreed]

Ian Pitts: I was apprehensive. I didn't see how it was going to work with the workflow, although a very dangerous workflow that we had kind of adapted at work. So he came in and said one day, "We really need to get this stuff on version control." And I think I kicked and screamed a little bit and then he held my hand and walked me through it. He might actually be better to give a quick rundown on kind of the nuts and bolts and a quick overview elevator pitch with version control. Adam?

Adam Wiggall: Okay, so as the name suggests, it's a way of controlling the version of the code that you are using for a website or any sort of software project.

Lea Alcantara: [Agreed]

Adam Wiggall: The software itself, Git, in our case, and there are the versions out there – although I've only ever worked in any depth with Git –It gives you a really streamlined way in a sufficient manner to keep lots of code and a history of code on your system.

Lea Alcantara: [Agreed]

Adam Wiggall: I think we've been all there where we have a file name such index old version 2 revision 12

Lea Alcantara: Yeah.

Adam Wiggall: And then if you are working with other people on a project, they will say, "Oh, we are at revision 12 old or are we at old revision 11.

Emily Lewis: [Laughs]

Lea Alcantara: Yeah.

Adam Wiggall: And you have a massive amount of files to keep track of on anywhere with projects, and it sort of makes that very simple. I think people shy away from it initially because traditionally it was a command line tool so you have to dig in to the terminal to use it and that's intimidating for a lot of people.   It's so intimidating for a lot of designers.

Lea Alcantara: Yes.

Adam Wiggall: But in recent months, there is a lot of good interfaces that helped with that learning curve, and yes, you can use it without going anywhere near the command line tool.   So it's got quite friendly in the last 12 months, I'd say. Software like Tower makes it extremely simple to getting involved with version control.

Lea Alcantara: Interesting. Well, what's interesting with your statement is this seems still a very young process.   The fact that it used to be limited to just the command line for several years, it really imposes a restriction for a lot of people who work on the web. Is this still constantly evolving? Are there any particular limitations to the software, or is there actually a preferable way to work with it, or is it just personal preference?

Adam Wiggall: I think that the version control itself has been around for a long time.

Lea Alcantara: [Agrees]

Adam Wiggall: And Git is certainly the version control software that we use.   It isn't new.

Lea Alcantara: Yeah.

Adam Wiggall: Most of the commands and the procedures have been around for a long time. I think that it's probably new to designers and front end people who are coming to more a programming environment than a design environment, but the software is stable. The software has been around for a while. I think it stayed, the GUI elements, with the software like Tower that's new to the scene, and although it's still a new software, it's resilient, it's stable. So there should be no reservations about getting involved because of any worries about changes or updates or anything like that.

Emily Lewis: So Adam, you've mentioned several times that it's something that you believe that is something new for front end developers and designers. But does it make sense as a tool for someone who is strictly doing front end development and design? I mean, does it still have a role in those workflows, or is it really lending itself to sort of back end development?

Adam Wiggall: No, definitely, if you are involved in front end, you are writing lots and lots of codes, whether it would be HTML, CSS, JavaScript. So having that all into version control is an absolute. It's a massive, massive plus to any project that you get involved with. I've even spoken to a few developers, designers, front end people, and freelance people. Often they work alone, and it's just them and the client, and that's one of their reasons for not getting involved. They say, "You know, it's just me. I don't work on a team. I know my codes are distributed.

Lea Alcantara: [Agrees]

Adam Wiggall: If you work on your own and you are not using Git, that's never an excuse because it really, really helps cut down so many of the problems you can have if you need to go backwards in your code, revert to something that you were looking at previously if at that time, they changes their mind on something. If you want a new feature or an add-on, having a version control system in place makes all of that so much easier. So whether you are on own or on a big team or even a small like we are, we are ready. We don't want to live without it. That's all in that.

Ian Pitts: Yeah.

Lea Alcantara: So…

Ian Pitts: I've worked on a number of side jobs, not part of Pitney Bowes, and even though I'm pretty much a one-man band on most of these sites, I still us Git to version control my code. One of the nice things is that it keeps a backup. It keeps a backup of all these different versions on your local machine. I pushed to a free service called Bitbucket on sites that I don't need to deploy and so that means I have another copy there. I have another copy on backups of my local machine. And one of the nicest things is many times, even as a front end designer, you come up with an idea or maybe a way to do a specific layout for like an e-commerce store front or something, and then later on, you think maybe you are trying to get to sleep at night or in the morning, you have this epiphany and you are like, "Man, I could totally do this another way."

Before version control, you would just start changing all your code and just trying it and seeing what happens, and if it works and it's better, then you won. If not, then you either have to go back to your backup files, which could be JavaScript, CSS and markup, and somehow undo those changes, but what you can do with Git is just branch your repository. Start making your changes in this branched copy. If it all works out, you just merge it back and you won. If it doesn't work out, you trash the branch and you are back where you left off without having to go and hunt around and change things.

Lea Alcantara: Before we move forward more about the technical stuff, we talked a little bit about maybe some of the fears, the barriers and to when you even get started. You mentioned Bitbucket, which was free, but are these version control solutions free, paid, affordable or expensive?

Adam Wiggall: Well, Git itself, and we keep talking about Git because we don't have any vast knowledge of any of the other systems, and Git is by far and away the most popular version control system in that realm at the moment.

Lea Alcantara: [Agrees]

Adam Wiggall: Git is a completely free piece of software.

Lea Alcantara: Okay.

Adam Wiggall: Whether you are on Windows or Mac or Ubuntu or Linux system, as you know, there are different versions for each and it's completely free. The repositories that we are talking about, assuming like with Bitbucket, GitHub as well, of course, Beanstalk and a few others have all varying levels of accounts. Most of them have free accounts, and then you have paid accounts for more enterprise-type requirements. But also you can have repositories in Dropbox, and so that's a free solution for a lot of people, and you could also set up a Git server, which is probably beyond the realms of most people and beyond their needs. So it's completely free, there is nothing to pay and you can put your code in an offsite repository for free as well with just a little bit of smarts about how you do that, probably through Dropbox system that links them up.

Emily Lewis: Now, in terms of how the two of you got started, Ian mentioned that at the outset that it was your idea, Adam, to sort of bring version control to the projects that you guys were doing for Pitney Bowes. How did you establish how you two as a team and perhaps with other people bring this into the environment at Pitney Bowes in the projects there?

Ian Pitts: Yeah, so the way we really adapted it, we had a new project come up that was a new development. It was actually a new free standing installation of ExpressionEngine, not hopping on of the arguably overloaded multi-site manager installs we have, and it was also ExpressionEngine 2.x, and both Adam and I were going to be collaborating heavily on it. The configuration, I don't want to get too much technical stuff here, but essentially, one of us did the installation of ExpressionEngine locally. I believe we both used MAP, which is the LAMP Stack for Macintosh since we are both Mac-based.

Lea Alcantara: [Agrees]

Ian Pitts: So someone did the installation, the initial installation, and installed ExpressionEngine, configured the database, got all of that set up and running and checked it into the local Git repository. There are some configuration changes that are needed, both the config PHP and database PHP that enable those files to be pre-configured for multiple environments so that the local environment variables, such as host name, username, password, and so on are stored in there as well as the production environment, but right at the onset, we pretty much just synced up. We used the same database name, the same username and password so that gets checked in. And then we push it out to a central Git repository. In our case for work, for Pitney Bowes, we used Beanstalk.

Lea Alcantara: Okay.

Ian Pitts: So then, let's say, if Adam started, he would tell me everything is up there. I would pull it down to my local repository and pop in a few and it create the database and execute the SQL script that's dumped in with all the rest of the files, and then boom, I have a working snapshot of everything that he had up to the last time he pushed it.

Adam Wiggall: It might be worthwhile if I jump in at this point. Some people might be listening and think, "Oh, where is all this going on?" whenever we run a Git or a version controlled site, we are always talking about running that site locally on our own machines, so that's done through a local host, completely offline if need be. That's when that comes in because it helps us set it up.

So we run a version of the site locally and everybody who is working on the project will have that version and then we will have a staging and/or production site, which will be out on the web somewhere. All these databases that Ian is talking about and all that, they all are existing either on our own machines so we can run our own local version of the site, or in the case for production or a staging server, it might be with EngineHosting or anybody like that, there will be a separate database out there so that there will be many versions of the same site being run at the same time throughout the team.

Emily Lewis: When you got this all set up, I mean, it sounds like a nice workflow, but I'm just curious, is this something that sort of documented with Git, or is this something, this workflow that you all customized based on what you needed for Pitney Bowes and/or what you had done in the past? I mean, is there some sort of getting started documentation to have kind of help someone get started?

Adam Wiggall: Well, I think credit where credit is due here. I got started in it because of Ryan Masuga's talk or the slides from Ryan Masuga's talk from EECI 2010 in Leiden.

Lea Alcantara: [Agrees]

Adam Wiggall: It's where Ryan made a comment in that presentation that not using Git is completely irresponsible and a disservice to your clients. So I don't anyone who is not using it to feel bad, but at the time when I heard that, I thought that was a little bit, possibly a little bit of an exaggeration, but I've got into it because of that comment. Ryan has got a great presentation. It's still online with PDFs. It's called *Let's Git It On.*

Lea Alcantara: Nice.

Adam Wiggall: Yeah. I picked up the initial as sort of way of doing things from there. I tried to document it. I think when Ian and I get together, I documented as much as I could about the practice that we try and follow, and then after a week or so of constantly IM-ing each other backwards and forwards with that. No, we didn't get it and things weren't working out, we started to laying out a process. So it helped us to definitely develop a style of using Git, but not only that, I think of collaborating on projects, it helped formalize, if you will, the way that we put stuff together and certainly, our directory structures and everything else have become a lot more streamlined and has been similar across projects because of this initial setup that we did.

Emily Lewis: And what's the bottom line for that? You mentioned that it's streamlining your processes and it's making it easier for you two to collaborate. Is that resulting in more efficient development? Are you getting projects done faster? Are you able to deliver faster?

Ian Pitts: I believe so, definitely. One of the challenges is we have two options. We can either develop live on a central staging server or central development server where there is one copy of the file, one copy at the database and each of us are just connecting in remotely and making template changes and making database changes. That's a little bit dangerous for obvious reasons and if you don't have a separate production and development environment, it's even more dangerous. So what we ended up doing, and ExpressionEngine does throw kind of wrench in things because a great deal of the functionality in ExpressionEngine is tied into the database.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Ian Pitts: Content is in the database.

Emily Lewis: Yeah, snippets.

Ian Pitts: Snippets are in the database. Global variables are in the database. All your custom field configuration, everything is in the database. As a result of that, we do have to modify our workflow a bit since everything is distributed and the database being in sync involves us doing a dump of the database via some kind of tool. I've used Navicat. I think Adam uses Sequel Pro. What that essentially does is it dumps the SQL file of the entire contents to the database and we put that in a known location that's actually checked into Git. The caveat is if one of us, let's say, needs some database changes to add some custom fields and the other one of us started adding content, one of us is going to lose our data. It's kind of pain in the pain and more work than it's worth to try and merge database changes. It can be done, but it's a pain. So we kind of have an informal hand off where one of us will get started on templating and maybe putting assets together while the other is building out custom fields and so on, and then at the next check in we might trade back and forth.

Emily Lewis: You have mentioned a number of tools that you are using. Is there anything specific not necessarily to Git but to ExpressionEngine that you found you've had to use like add-ons or anything to accommodate working with ExpressionEngine in this manner?

Adam Wiggall: I think a couple of the add-ons that we use, we use simply to pull information stored in the DB and add into the file system. We always add templates as files and I think the bulk of the community has been doing that for a long time in there. But we also pull out snippets and global variables, so it's there in the file system as well. In that way we can quarantine, if you will, the changes to DB that needs to be done. If we are changing snippets or global variables, we don't need to edit the database to do that. Those add-ons will be linked I'm sure in the show notes as to what we used there, but it's very minor. Navicat is probably the biggest piece of expense that we've had to go through in order to do this, and we use Navicat because it's got HTTP tunneling on it. The bulk of the sites that we work on are EngineHosting and we don't have any other way of connecting to those other than through HTTP tunneling, so Navicat is the only tool out there, or it's not the only tool, but it's the most reliable tool that we found for using HTTP tunneling to make changes to the database remotely.

Lea Alcantara: So can you briefly explain what HTTP tunneling is?

Adam Wiggall: From a technical point of view, I haven't got a clue.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Adam Wiggall: From sitting here and working with it point of view, basically if you can connect to a server through what's called SSH, a secure shell environment.

Lea Alcantara: Yes, yeah.

Adam Wiggall: You can pass codes in from your local machine through SSH. We don't have that facility with the bulk of our hosting, so we basically go through the HTTP protocol in order to affect changes to the database, so with Navicat, it has a built-in HTTP file that you upload to your server. It's so secure. It's safe and sound, and it just if you like creates a tunnel along the normal HTTP channel as opposed to an SSH channel which is the traditional way of connecting in.

Emily Lewis: I'm looking at some of the notes that you guys provided before we got started on the podcast today, and I'm just curious that you mentioned use of Securit:ee and how you are using that to restrict channels to – it looks like a deployed site.

Ian Pitts: Yeah, that's actually a pretty funny one. So we identified this potential issue with websites that are currently deployed and allow content owners to add content. The challenge is if the site is live and you need to make some changes. You need to make sure that that database doesn't change, and one option is email all of the content owners and let them know, "Over the next week, we are going to be making some changes to the site, a new development that you asked for, but we need to have you not log in and make any changes at all."

The problem is there is no way to actually physically lock them out. You can tell them all day and they might still do it or soon will forget. So having that, Eric at EECI and I knew he had this product called Securit:ee. I asked him how hard he thought it would be to add a feature that would enable a super user to lock out all other member groups in the control panel. A few days later, he had implemented it and after one or two rounds of back and forth, a little bug testing, it was ready to go and it works like a charm.

Lea Alcantara: That is such a useful add-on. I wish I had that back when I upgraded one of my client sites from EE 1 to EE 2 and I had to do a whole bunch of things, and I told them, "Don't touch the database," and inevitably, my client did, [laughs] and added a new entry. But fortunately for me, it was only one thing and I knew where to change it, so when everything moved forward I just made that. I just duplicated what they did, but I can't imagine if there were multiple content managers out there and then making several different changes or they missed the email or it went to spam or whatever, having to manually deal with that would be a pain.

Ian Pitts: Yeah, definitely.

Emily Lewis: I'm just curious, this sort of goes back to – Adam, you've mentioned a couple of times that your experiences with Git, is that just because that's the one you picked up, or did you compare it to something else at some point and decided that Git was what you wanted to move forward with?

Adam Wiggall: Well, I picked it up initially because other people in the community were already using it and it was, therefore to me, it was tried and tested.

Lea Alcantara: [Agrees]

Adam Wiggall: When a guy like Ryan Masuga is willing to stand on stage and put his name behind it, that's enough for me. You don't have to be around too long in the version control sort of forums and all the rest of it to see a lot of people praising Git over the other big one which is SVN. And if there is any SVN fan boys in your listeners, they might disagree with this. But from what I heard, Git was far superior, a lot more lightweight with ways you have to branch and branching really is one of the killer features of Git. It really is just the branch, and so I never looked at anything else. I'm a lot like most people there, once they find it, they never look at anything else from that point onwards and Git is pretty similar.

Emily Lewis: In terms of how you guys have developed this workflow, have you brought any other team members or any other developers into a project that you were using this?

Ian Pitts: I'm trying to think. We have slowly started expanding to other members of the team, but I don't think we have yet had a situation where we have four people all editing code on the same project. We have had an instance just between Adam and I, and this is a little bit different. We did have an instance where we had to build out a site that was quite similar to the previous site. So being able to clone the existing Git repository pretty much gave us – I don't even know how many hours we saved – a big time savings by already having ExpressionEngine installed, configured, custom field groups created, templates created. The site actually shared a lot of the design from the previous site as well, so we were literally able to roll out this other site barring waiting months on content in very short order.

Adam Wiggall: We've got a new project set up at the moment or other projects that are going on within the ecosystem at Pitney Bowes that we have a new team member on and working and making commits to. She has a design background and certainly a very front end web development kind of background. Ian, you may be more to say on this side. But I think that it's all going well for them and that she's pitching in to that new process quite well.

Lea Alcantara: Cool.

Ian Pitts: Yeah. Yeah, we actually have two other members on the team. One who is more focused on front end. The other that's definitely more focused on back end, but really wasn't versed in any version control, and they both have been able to pick up on it pretty quickly, especially given the short amount of time they've been both exposed.

Emily Lewis: Now, when you are working with that many people, and this is my ignorance with how Git works in version control in general, is it literally like a check in, check out sort of thing, or is it you are working on something and someone else could be working on the same thing and then they get merged?

Adam Wiggall: Yeah, I mean, there are several different workflow techniques you can use, but you basically all have a copy of the code. It's a distributed system. So you've all got your own copy of the code and then you nominate a repository as sort of the old master repository, which isn't enforced by Git. It's just something that's become convention. And when you've made your changes and you are happy with them, you push that up to the master repository if you will, and then people can pull down those changes. If there is anything in those changes that clashes with the changes that you've got, Git will tell you that and point you to the places in the file that you need to address. In the Tower, in like the GUI, the Tower will walk you through that quite comfortably. If you are on the command line, it can be a little bit more painful, but everything will be pointed out to you that crashes. But where it can, it will work out your changes, and in effect, my changes, and it will put the two together and everyone can live peacefully side by side without having to constantly communicate with other about what they are going to be editing. And I know a lot of small teams are constantly backwards and forwards moving files saying, “Did you update it?” “Yes.” “Great. Well, can you copy that to my drive, or can you copy that to the network drive or something like that, so that I've got a new copy?” And just working like that now seems archaic to us and fraught with danger.

Emily Lewis: Ian, you had mentioned that you've used Git a bit with some one off-project you are doing on your own. Has it been the same kind of experience in terms of how quickly and efficiently you can get work done on just like a single-person project?

Ian Pitts: Yeah, definitely. I haven't been keeping up with kind of my start point repository as much as I should, and right now I have four different sites going on and I kind of lose the ability to have one site that has everything configured how I love it with matrices for content that handle images and then some replacement plugins that let me place keywords to put server side, CE image, resized images within body content and so on. But that's kind of the idea, so Adam and I created what we call Start Point that has very generically named field groups for static content. It has structure. It has a lot of the plugins and add-ons that we would generally use for a project. It might have a news channel preconfigured with proper fields, maybe a calendar channel, testimonials channel and all of that's ready to go. So when it's time to start a new project, we clone that repository and then customize it, which cuts out a lot of the initial development.

Emily Lewis: Yeah, I mean, it seems to me that that's a fantastic reason just to use it regardless of anything else, and just having something already set up that you can branch off of. Is that what you are doing?

Adam Wiggall: Well…

Ian Pitts: In this case… oh, I'm sorry, go ahead, Adam.

Adam Wiggall: Well, yeah, in this case, you would be cloning what you've already got.

Emily Lewis: Okay.

Adam Wiggall: But branching, cloning, those phrases are all – you are all doing pretty much the same thing. You can choose to clone a repository and you get a carbon copy of it, or you can go into a repository and branch it. But you would probably only do that if you are looking to put a new features or bug fixing or stuff like that.

Ian Pitts: Yeah, a great example of where we would branch with something, let's say, that Start Point had ExpressionEngine 2.1.6 and the latest and greatest came out two weeks ago, the early adapters had already dug in and did their bug fixing in the latest EE build. We would go and branch our Start Point repository, install the new version of ExpressionEngine, test it to make sure everything seems kosher and then merge that branch back in.

Lea Alcantara: Is there anything else you would like to tell our ExpressionEngine listeners about version control or Git, specifically with ExpressionEngine? Is there any advice that you would like to give for those that want to try this?

Adam Wiggall: I'll start on that. First of all, it may seem like a lot of work to learn, but initially, that hump is worth getting over, and also to get over that, there is a lot of resources already in the community with some great work by Eric Reagan and also Matt Weinberg.

Lea Alcantara: [Agrees]

Adam Wiggall: They've all published articles or put a code up to use a copy and use which helps you set up the environment correctly to get you well on your way. And my parting shot would be once you start using it, you will never ever, ever go back.

Emily Lewis: [Laughs]

Ian Pitts: Yeah, I have to agree with that. As I said in the beginning of the podcast that I was initially resistant and with some stuff to learn and there was that scary command line sometimes. But in the end, I won't go back. I mean, Git is preinstalled in my Mac. I can use it from the command line, and on every single project that I create, whether it's a personal project or whether it's for a client, I always have Git managing every aspect.

Lea Alcantara: Pretty cool, pretty cool. All right, I think that's all the time we have for today. Thank you, Ian and Adam, for joining us and sharing your experiences.

Adam Wiggall: Thank you, Lea.

Ian Pitts: Thank you for having us.

Emily Lewis: And in case our listeners want to follow up with you guys, where can they find you online, Ian?

Ian Pitts: Oh, let's see, I am on Twitter @iso100 or you can get me via email at iso100@gmail.com.

Emily Lewis: And you Adam?

Adam Wiggall: I am @turnandface on Twitter, so that's the very best place to get a hold of me.

Emily Lewis: Great, thanks again, you guys.

[Music]

Lea Alcantara: Now, we would like to thank our sponsors for this podcast, mithra62 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, ee-podcast.com.

Emily Lewis: And don't forget to tune in next time when we have Lisa West from EllisLab giving us the lowdown on official ExpressionEngine support and an update on what EllisLab is up to. That will be on Thursday, February 9th. Also, be sure to check out our schedule on our site, ee-podcast.com/schedule for more upcoming topics.

Lea Alcantara: This is Lea Alcantara.

Emily Lewis: And Emily Lewis.

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

Emily Lewis: Cheers.

[Music stops]