



Under the Scope

Stacking Up Middleware
Serializing PHP Objects
Defining Project Metrics

ALSO INSIDE

Education Station:
Autoloading Your Code

Community Corner:
PHP 8 Release
Managers:
Interview with Sara
Golemon and Gabriel
Caruso, Part Three

The Workshop:
Mail, Ngrok, and
Xdebug in WSL

Sustainable PHP:
Build a Composite

Security Corner:
Observable Security

PHP Puzzles:
Calculating Population
Growth

finally{}:
We Are All
Programmers



Free
Sample
Article

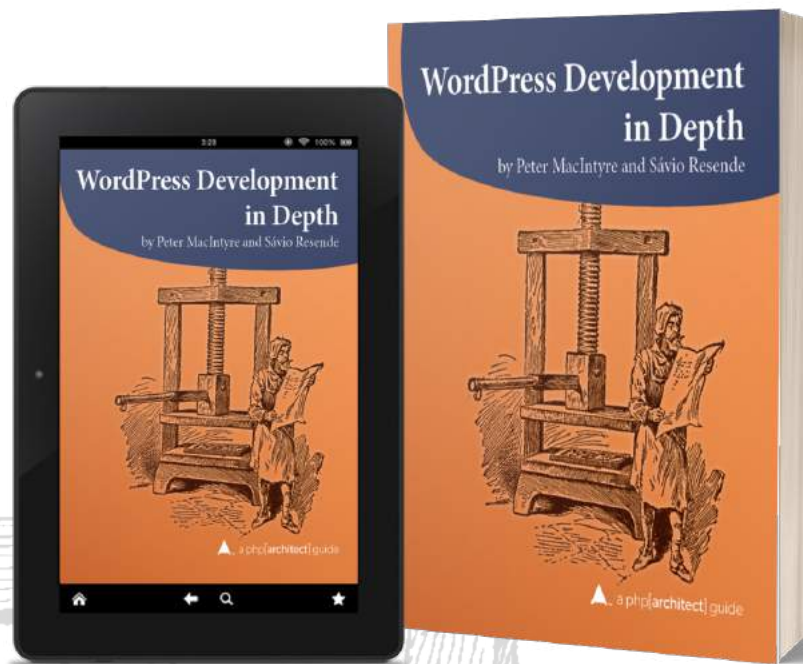


Learn how a Grumpy Programmer approaches testing PHP applications, covering both the technical and core skills you need to learn in order to make testing just a thing you do instead of a thing you struggle with.

The Grumpy Programmer's Guide To Testing PHP Applications by Chris Hartjes (@grmpyprogrammer) provides help for developers who are looking to become more test-centric and reap the benefits of automated testing and related tooling like static analysis and automation.

Available in Print+Digital and Digital Editions.

Order Your Copy
phpa.me/grumpy-testing-book



Build a custom, secure, multilingual website with WordPress

WordPress is more than a blogging platform—it powers one-fifth of all sites. Since its release, enthusiastic contributors have pushed the envelope to use it as a platform for building specialized social networks, e-commerce storefronts, business sites, and more. This guide helps you sort through the vast number of plugins available to find the ones that work for you. You'll also learn how to use CSS and PHP code to tailor WordPress to your needs further.

Written by PHP professionals Peter MacIntyre and Sávio Resende, this book distills their experience building online solutions for other WordPress site builders, developers, and designers to leverage and get up to speed quickly.

Read a sample and purchase your copy at the link below.

Order Your Copy
<http://phpa.me/wpdevdepth-book>



PHP 8 Release Managers: Interview with Sara Golemon and Gabriel Caruso, Part Three

Eric Van Johnson

Part three concludes my interview with the PHP 8 Release Managers about PHP Internals. We touch on getting started contributing to internals via RFCs, becoming release managers, the commitment required by that role, what happened to PHP 6.0, and how the project's release cycles have evolved.

Being a Release Manager

I have to be honest Gabriel, you really surprised me with the fact that you've only been coding in PHP for four years and here you are the Release Manager of PHP 8, can you talk about that journey a little?

Gabriel: When I finished school, I was familiar with PHP. I had an understanding of what PHP could do. I honestly don't remember why but I had the idea to look at the source code, so I went looking for it because I was learning about Open Source and starting to understand the meaning behind Open Source. When I found the source code, downloaded it, compiled it, I then made a change and contributed it back, and it was accepted. I was amazed that I could actually change the language.

The first time I contributed to internals and proposed a change, I was asked one question, "why?" You see the real impact when you want to make a coding change to your internal application. How many people are actually impacted by that change? A thousand? A hundred? A hundred thousand? Maybe a million if you have a framework. But when we are talking about making a change to PHP's source code, I couldn't tell how many developers, how many projects, how many businesses we are going to touch. So the question "Why" is such an important one. Give me a reason, give me arguments, give me numbers, give me a theory, give me other examples. Because you have

to remember, you are not changing a project. You are changing a language.

I realized I needed justification for my request. I needed to get the arguments for why I wanted the change I wanted, and I needed to do the research to justify it.

My first RFC was for a small function back in PHP 7. Someone replied, "OK, this is a nice function, and you have nice arguments, but why? Why does this need to be in the core and can not just live in a library?" That triggered me to research my request more. I asked myself, why should this be in the core.

We have contributors today that specialize in researching the impact of their RFC on the language as a whole, and I am pushing myself to become more thorough. They do things like download 10,000 of the top packages from Packagist and run analysis on them to figure out how many of them would be impacted by the change they are proposing.

Something PHP internals taught me was how to research the impact of a feature, which has helped me with my professional career.

I like to let people know that if there is something about PHP you don't like, look at the source code and see how you can make it better. You don't have to be a C++ developer to understand a lot of the code. I'm not. I wasn't back then, and I'm still not, but I learned enough to make the change I was looking to make. If you want, you can change PHP.

Later, I saw Derick's email asking if anyone wanted to be a Release Manager, and I noticed that not a lot of people were replying to it. I wondered why that was. I thought that it couldn't be that hard to be a Release Manager, and I decided to toss my name into the hat and said: "I want to be the Release Manager". I honestly didn't think I had much of a chance and almost immediately thought I had just made the biggest mistake of my life.

After the first vote, I was tied with Ben Ramsey. I thought, "He should be the release manager; after all, he is the guy behind UUIDs," and I am still unsure why people started voting for me. When I won, I was really worried, but then I remembered how I had gotten to that point. If I didn't know something, I just asked, and I remembered how much support I got. I started thinking, "I can do this."

Sara: We try not to let Release Managers cover multiple releases at the same time, and we try to partner veteran Release Managers up with newer ones. Since I was the Release Manager of PHP 7.2, I am coming in just under the wire because PHP 7.2 will be End Of Life before PHP 8.0 is released.

So, a Release Managers's job isn't over when the version of PHP is released?

Sara: It's roughly a three and a half year commitment with the first half-year being the pre-release cycle such as Alphas, then GA, until EOL 3 years later.

Release Cycles

Can you explain the changes in the release cycle between versions? I remember the move from PHP 4 to PHP 5 being a prolonged process, then the different versions of PHP 5.x all seemed like slow releases, I won't even ask about PHP 6, then we hit PHP 7 and all of a sudden we are getting regular releases on a pretty quick pace, what all changed?

Sara: Oh, I'll talk your ear off about PHP 6 if you want, and believe it or not, PHP 6 is part of this. The shift actually started before PHP 7, but it was PHP 7 when we really kicked it into high gear. For example, PHP 5.6 was already on that 3-year release, and we will have PHP 7.0 come out next year. Back to PHP 6, it was actually part of this because if you remember, the 5.3 release came after a long drought in the PHP development cycle. That long drought was in part because of the work we were doing on PHP 6.0. We had this whole branch where we were adding in Unicode, and we were making some great progress, but we were also learning things along the way, including things like maybe we don't want 16-bit characters all over the engine. We were also doing a lot of feature work like Namespaces, Closures, and other things that would eventually end up in PHP 5.3.

When we decided to kill PHP 6.0, we were left with this big dump of features that became PHP 5.3, but it was also kind of a come to Jesus moment of, "we need to think about what a 'release' is going to be." The fact that there are now books out there talking all about PHP 6.0, we knew we needed a better plan for how we were going to approach our versioning system, our release process, and how we were going to define what features are going to be added. That's the conversation that grew into what became our RFC process and our release process.

As a pure open source project, the idea of introducing processes felt very enterprisey and very gross. However, I think nobody who was around at that time and is still actively working on

PHP would deny that we are so much better off in terms of understanding where we are and where we're going because of those processes. And yes, that means that we now have a release every single year. And then, after three years, we stop supporting that release. We can add to that a couple of positives. Number one, the difference between two versions has become much, much smaller. The upgrade from 5.2 to 5.3 was, in addition to getting a bunch of new features, it was also really painful because a lot of time had gone into these changes that had happened. We also had some potential unexpected changes because of bits that got left-over from the Unicode work. By having a release every year, we can just say clearly and definitively: this is what's changed, these are the features, and these are the possible breaks. We can have very clear upgrading guides that tell you here's what you might need to worry about, here's what you don't. The upgrade process as a whole should be much smoother.

I do think PHP should pick up the concept of an LTS. I don't know exactly where an LTS would fit. Maybe it would be the last version of a major. So like 7.4 could be an LTS in the way that 5.6 got slightly extended support. Perhaps it's every third release that could be. I don't know. The trouble is that then becomes a higher commitment from the Release Managers, and instead of three and a half years of commitment, you now

have five or seven years of commitment from a pair of individuals, and that's asking a lot. That is something that we have to give some very careful thought to before we just jump into it.

Gabriel, being the new PHP Release Manager, and PHP 8 getting close, how is the pressure? Would you recommend to others to become a Release Manager in the future?

Gabriel: This is actually fun. Many people ask me, "What's it like contributing to PHP Core?" or "What's the experience been like being a Release Manager?" It's fun, it's nice, and I am learning a lot. I've gotten to meet some great people like Sarah, Nikita, and Derek. I've spoken at conferences, I'm learning, and I am just enjoying myself. It doesn't keep me up at night.

I watched the launch of Space X, and I thought coding for that would keep me up at night wondering what would happen if I introduced a bug. But contributing to PHP, we have tests, reviews, Github, we have documentation, and we have a committee. We have a bunch of stuff.

Contributing to PHP Core has changed my life, and every day I am learning.

Sara, as the wily vet, is Gabriel naive?

Sara: He's not naive at all. I wouldn't still be doing this if I weren't getting something out of it, and often that something is learning something unexpected. There's always more to learn.



Eric Van Johnson is the CTO of DiegoDev Group, LLC. A group of passionate and talented frontend, backend, and mobile developers that strive to provide outstanding services. He is also one of the organizers of San Diego PHP (SDPHP), his local user group, and a podcaster. In the early '80s while Eric's friends were getting Atari 2600 or Intellivision gaming consoles, his Dad bought him a TRS-80 Coco with BASIC. Eric started teaching himself coding and never looked back. A husband, father, and enjoyer of scotch and baseball. He had to give up playing baseball because he kept spilling his scotch while trying to run the bases. @shocm



Borrowed this magazine?

Get php[architect] delivered to your doorstep or digitally every month!

Each issue of php[architect] magazine focuses on an important topic that PHP developers face every day.

We cover topics such as frameworks, security, ecommerce, databases, scalability, migration, API integration, devops, cloud services, business development, content management systems, and the PHP community.



Digital and Print+Digital
Subscriptions
Starting at \$49/Year

http://phpa.me/mag_subscribe