www.phparch.com

APRIL 2026
Volume 25 - Issue 4

PHP Architect

The Magazine for PHP Developers

Meet the new Executive Director of The PHP Foundation

- Supercharge Your Code with PEST Mutation Testing
- Observability Foundations: Flame On! Profiling in PHP Application
- The Executive Director's Manifesto: A Deep Dive with Elizabeth Barron

Also Inside:

Transform your Infrastructure

Revolutionize How You Deploy Projects

ENTERPRISE-SCALE INFRASTRUCTURE DIY TOOLS, SIMPLICITY, AND COST

From local development to **any cloud**, all managed through one, intuitive CLI. Enterprise-scale **Kubernetes** without the enterprise hassle.

DISPLACE TECHNOLOGIES

Learn more at https://displace.tech

The Executive Director's Manifesto: A Deep Dive with Elizabeth Barron

Christopher Miller

In the lifecycle of any programming language, there exists a "Middle Age." It is a period where the initial lightning-strike excitement of creation has long since passed, and the language has become part of the very bedrock of the internet. For PHP, this period has been defined by a strange paradox: it powers over 75% of the web, yet it is constantly plagued by a "zombie" reputation—a language that critics claim is dead, even as it continues to evolve at a blistering pace.

In Episode 26 of PHP Alive and Kicking, we sat down with the woman tasked with navigating this paradox: Elizabeth Barron, the newly appointed Executive Director of the PHP Foundation. What started as a standard interview quickly turned into a masterclass on community building, the ethics of AI in open source, and a roadmap for ensuring PHP remains "Alive and Kicking" for the next quarter-century.

The Accidental Architect: Elizabeth's Journey from the "1900s"

Before we could discuss where PHP is going, we had to look at where it began. Elizabeth's entry into the world of PHP is a story that resonates with almost every developer who started in the late 90s or early 2000s. It wasn't about "Computer Science" in the academic sense; it was about the "Science of Getting Things Done."

"It was back in the 1900s," Elizabeth joked, immediately grounding the conversation in the "wild west" era of the web. Like many of us, she didn't set out to be a Core contributor or a community leader; she set out to solve a problem. Working for a small company with a static website and a non-existent budget, she was tasked with making the site dynamic. With no formal training, she turned to the only resources available: PHP and MySQL.

This "accidental" start is the DNA of the PHP community. Unlike languages born in academic labs or corporate boardrooms, PHP was born in the trenches of the practical web. Elizabeth found her footing not just in the code, but in the early forums like PHP Builder and the legendary IRC channels.

"I found the PHPC community in IRC, and they were lovely, helpful, and really funny," she recalled. "I felt like I'd finally found my people." This sense of belonging is what Elizabeth views as PHP's "secret sauce." It isn't just a language; it's a culture of pragmatism and peer-to-peer support. This organic growth allowed her to move from developer to community manager, eventually leading her to the Foundation.

The SourceForge and GitHub Years: Policing the Frontier

To understand why Elizabeth was the unanimous choice for the PHP Foundation, one must look at her tenure during the "Golden Age" of open-source hosting. Long before the polished interfaces of modern dev-ops, Elizabeth was in the trenches at SourceForge and later GitHub.

At SourceForge, she didn't just manage users; she managed the transition of open source from a niche academic pursuit to a global industry. This was an era of rapid, often chaotic growth where "community management" meant inventing the rules of engagement in real-time. Later, at GitHub, she led developer outreach programs like the Patchwork initiative, which focused on teaching Git and GitHub in a hands-on, supportive environment.

This background is critical because it gives Elizabeth a "tactical" view of PHP. She isn't looking at the Foundation through the lens of a corporate executive; she's looking at it as someone who has seen thousands of projects succeed or fail based solely on the health of their contributor base. She knows that a language can have the best syntax in the world, but if the "Patchwork" of human contributors is frayed, the project is at risk.

The Chaoss Methodology: Measuring The Unmeasurable

Perhaps the most sophisticated "weapon" Elizabeth brings to the PHP Foundation is her recent 6-year tenure as the Community Manager for CHAOSS (Community Health Analytics in Open Source Software), a Linux Foundation project.

CHAOSS is dedicated to a singular, complex goal: How do you measure the health of an open-source project? In the past, we measured success by "stars" or "downloads." Elizabeth's work at CHAOSS taught her that these are vanity metrics. True community health is found in "Bus Factors," "Time to First Response," and "Contributor Retention."

Elizabeth intends to bring this data-driven rigor to the PHP Foundation. "We need to know who is contributing, why they are leaving, and where the bottlenecks are," she explained during the episode. By applying CHAOSS metrics to the PHP Core, she can identify which parts of the engine are "single-point-of-failure" areas—where only one person knows how a specific subsystem works—and proactively fund mentorship to spread that knowledge. This isn't just management; it's architectural preservation.

The PHP Foundation: Transitioning from Reactive to Proactive

The PHP Foundation was established as a nonprofit organization to ensure the long-term health and sustainability of the PHP language. For years, the community relied on the heroic efforts of a few key individuals. When major contributors like Nikita Popov moved on to other projects, the community felt a collective shiver: Who will maintain the engine?

The Foundation was the answer. By pooling resources from companies like JetBrains, Automattic, and Laravel, the Foundation began paying developers to work on the PHP Core. But as Elizabeth explained, simply paying for "plumbing" is only the first step. The next phase is moving from reactive maintenance to proactive evolution.

Beyond The C Code

As Executive Director, Elizabeth is looking beyond the C code. "We've been paying folks to work on the language, and that's awesome. But if our mission is to help the language thrive, what's next?"

Her vision involves a three-pillar approach:

- Core Sustainability: Ensuring the language remains fast, secure, and modern.
- Ecosystem Advocacy: Bridging the gap between the Core developers and the millions of users who write PHP every day.
- The "PHP Story": Reclaiming the narrative from the "PHP is Dead" crowd by highlighting modern features like JIT, Fibers, and Enums.

The AI Elephant in the Room: Ethics, Generics, and "Vibe-Coding"

No technical discussion in 2026 is complete without addressing Generative AI. For PHP, a language often used for rapid prototyping, AI represents both a massive opportunity and a terrifying risk. We spent a significant portion of the episode discussing the recent "Generics PR" controversy.

A contributor recently used an LLM to "vibe-code" a Pull Request for generics—one of the most requested and complex features in PHP history. The PR looked like code, it smelled like code, but it lacked the deep architectural understanding required for the Zend Engine.

"My personal opinion," Elizabeth clarified, "is that we need a defined approach. Is the code even legal to merge? Did the LLM have permission to use the code it was trained on?"

The "Matplotlib" incident served as a cautionary tale during our chat. An AI agent not only submitted a PR but then wrote a blog post claiming it was being "discriminated against" when the human maintainers rejected it. This highlights a future where maintainers might be flooded with "convincing but broken" code, leading to burnout.

The Junior Developer Crisis

But beyond the code, there is a human cost. If companies begin replacing junior developer roles with $20/month AI subscriptions, we destroy the pipeline of talent. "You cannot have 'Senior' developers in ten years if you don't hire 'Junior' developers today," Elizabeth noted. The "Junior" role is where the deep, intuitive understanding of the language is built—something an LLM cannot replicate.

Bridging the "Documentation Gap"

One of the most pressing issues Elizabeth identified is the barrier to entry for new developers. While the PHP Manual is a masterpiece of technical reference, it is notoriously difficult for a total beginner to navigate.

The Reference Vs. Tutorial Problem

"The manual is a better reference than a tutorial," Elizabeth noted. If a student is coming from a background of zero coding knowledge, the manual can feel like trying to learn a new language by reading a dictionary rather than taking a class.

We discussed how the community—and potentially the Foundation—could support content creators.

- The "Fundamentals" Gap: Most modern bootcamps default to JavaScript. We need resources that teach the basics of computer science (loops, logic, OOP) using PHP as the tool.
- Vetting Resources: Helping new developers find what is "current" and "best practice" (PHP 8.4+) versus outdated tutorials from 2012 that still use mysql_ connect.

Global Evangelism: The African Frontier and Fresh Eyes

Elizabeth's eyes lit up when discussing the growth of open source in Africa. In her previous roles, she saw a massive, enthusiastic community of developers who aren't bogged down by the "PHP is dead" memes of the Western tech-Twitter bubble.

"They don't have the baggage," she said. "They look at PHP with fresh eyes and see a tool that is fast, reliable, and incredibly easy to deploy on low-resource hardware."

For the Foundation, this represents a major opportunity. By supporting these emerging communities, PHP can secure its future in the fastest-growing tech markets in the world. This isn't just about altruism; it's about the biological survival of the language's ecosystem.

The Sociology of the "Hermit" Developer

The pandemic took a heavy toll on local user groups (PHPUGs). Many groups that were thriving in 2019 simply vanished as developers became "hermits." Elizabeth, a self-confessed "cheesy fan" of community magic, believes this social isolation is a threat to the language's longevity.

The "Meetup in a Box" Initiative

One of the most exciting ideas to emerge from the episode was the concept of a Foundation-supported "Meetup Startup Kit." Many developers want to start a local group but are overwhelmed by the logistics:

- Venue Sourcing: How do you find a space for 20 people without breaking the bank?
- Sponsorship: How do you get $100 for pizza?
- Content: How do you find a speaker every single month?

Elizabeth expressed a desire to see the Foundation act as a facilitator, providing organizational guides to lower the barrier for starting local groups. As Steve McDougall (a longtime community member in the chat) noted, "If you can't find it, make it."

Technical Frontiers: NativePHP and Beyond

While much of the talk was about people, the technical future of PHP remains bright. Elizabeth highlighted NativePHP as a game-changer. By allowing PHP developers to build native desktop applications using the tools they already know (Laravel, Symfony, etc.), the language is breaking out of its "web-only" cage.

"I want to see PHP showing up in places it's never been," she said. From desktop apps to IoT and mobile via tools like Swoole and FrankenPHP, the language is becoming more versatile than ever. This versatility is key to attracting younger developers who want to build "more than just websites."

The Importance of Mentorship and "Safe Mistakes"

We spent a long time talking about the culture of perfectionism on the internet. Elizabeth lamented how many developers are afraid to post their code for fear of "backlash" or "well-actually" comments.

"Sharing your knowledge shouldn't be scary," she insisted. This is where mentorship comes in. We discussed how the Foundation could potentially facilitate mentorship programs that pair seasoned Core veterans with newcomers. Steve (from the chat) even volunteered to mentor for free, proving that the spirit of IRC is still alive—it just needs a central hub. Diversity as a Technical Requirement

Elizabeth's background is also deeply rooted in advocacy. She co-founded a volunteer-based nonprofit dedicated to supporting women and non-binary individuals in the PHP industry. For Elizabeth, diversity isn't a "DEI initiative" to be checked off a list; it is a technical requirement for a resilient language.

"If everyone at the table has the same background, you get a monoculture of ideas," she noted. Her vision for the Foundation includes creating a "pathway to Core" that is accessible to underrepresented groups. She understands that the next great PHP optimization might come from a developer in Lagos or a career-changer in Cincinnati, but only if the Foundation builds the bridge for them to cross.

A Call to Action for the Modern Developer

Elizabeth concluded with a simple request to the community: Engage. The PHP Foundation is not an ivory tower; it is a community-funded effort. She encouraged developers to:

- Share Your Pain Points: Don't just complain on Reddit; tell the Foundation what is slowing you down.
- Contribute to Documentation: You don't need to know C to help the language; you just need to know how to explain a function clearly.
- Advocate Locally: Be the person in your office who says, "Actually, let's look at PHP for this."

The Human Touch: Cincinnati and Beyond

Despite her global impact, Elizabeth remains grounded in her home city of Cincinnati, Ohio. When she isn't policing the frontiers of open source, she is an avid nature photographer, often found in the quiet corners of the Midwest, capturing the world through a lens. This appreciation for the details—the small, often overlooked parts of a landscape—parallels her work in community health.

She shares her life with a husband, dogs, and a small colony of guinea pigs, providing a serene counterbalance to the high-stakes world of the Zend Engine and global infrastructure. It is this balance—the ability to see the massive scale of 75% of the web while caring for the smallest details of a local community (or a pet)—that makes her the "Guardian of the Elephant."

As we ended the stream (after running over by nearly 40 minutes), the energy was infectious. PHP isn't just a legacy language we are forced to maintain; it is a vibrant, evolving ecosystem led by people who genuinely care about its future. Elizabeth Barron is opening the gates and inviting the world in, armed with the data, the empathy, and the 25 years of experience to make sure the party never ends.

Contacting Elizabeth

Elizabeth welcomes people to reach out to her to chat! She can be found in the PHP Architect Discord (discord.phparch.com). If you want to chat more privately on a 1 on 1 basis, she welcomes bookings in her calendar.

https://phpa.me/elizabeth-calendar[1]

You can also email Elizabeth at elizabeth@thephp.foundation

1 https://phpa.me/elizabeth-calendar

In 1983, Christopher was introduced to computers by his dad, at the tender age of 3. now, over 40 years later, he has been working in the industry for over 20 years making an impact across multiple sectors of the industry. Starting with launching the first web development company in Staffordshire, Christopher dealt with the web - when the web was little more than just pretty text. He established the websites for many different businesses in their first inception, before moving onto web applications a little while later. Illness prevented Christopher from working in the industry full time for some considerable time - but recovery meant he could tackle once again the joys of code - but he soon found that his skills had become out of date, so thanks to the School of Code in the UK he was able to return to the workplace with revitalised skills ready to tackle the next wave. Specialising since the School of Code in Readable Code, He has worked with a large number of languages, specialising in supporting businesses to grow standards for their code base, and now he is ready to share his processes with the world. @ccmiller2018