Why Is GitHub Taking Over the World?

Github has becoming the 83rd busiest website in the entire planet. That's a mind blowing statistic when you consider it's "just source control." Why is this happening?

by
Erik Dietrich
.
Jun. 10, 16 · DevOps Zone

There's a word out there that you'd be a lot more likely to hear from journalists, pundits, and authors than you would from techies: zeitgeist. The dictionary definition of the word, "the general beliefs, ideas, and spirit of a time and place," is straightforward enough. But when used by the people I mentioned, zeitgeist, as a descriptor, imparts *gravity* on the thing it describes. If I say that GitHub captures the zeitgeist of the programmer world of the 2010s, I am thus saying that GitHub is at the absolute core of the software development universe.

And I do say that.

If you're not familiar with GitHub, I'll offer a brief description. It is a website that wraps a software version control system called "Git" and allows software developers to host their code online for free (though there are paid models available). If you're a software developer, GitHub is a repository for you to store, exchange, trade, and talk about code. This may not sound like the stuff of which zeitgeists are made.

The sheer, raw popularity of GitHub alone doesn't explain this classification, either, though it is popular. Alexa, a web analytics company (not to be confused with Amazon's Echo personality), ranks GitHub as the 83rd most popular site on the planet, as of the time of this writing. That's particularly impressive when you stop to consider that GitHub's audience is the slice of software developers that feel like freely exchanging source code with one another. Other popular sites occupying top rankings, like, Facebook, Amazon, or Google, on the other hand, have an audience of literally everyone on Earth.

It isn't the improbable, raw volume of traffic to GitHub that elevates it to Zeitgeist level, however. It isn't even the growth or the market cap of the company or the number of contributing developers. Those things alone can't explain why GitHub is a zeitgeist — why it's taking over the world. To do that, let's look at it in a bit more detail.

From Improbable Beginnings

Let's take a little known, somewhat pedantic approach to storing code, wrap it in a cute user interface package, sprinkle a little social media on it, compete with an entrenched market dominator (sourceforge), give it away largely for free and change the world!

If I could go back to around 2008, when GitHub was just starting out, I can't say I'd be clamoring at the bit to become an investor. At least, not without 20/20 hindsight. It doesn't really sound like a game-changer, and yet that's what it became.

And yet, if you look at tech trends, the makings were there, if subtle.

Distributed Version Control for a Remote World

If you've been writing code for a long time, you no doubt remember the bad old days of remote work when it comes to source code version control. At the time GitHub first started attracting notice, centralized version control schemes were the standard, and when you were somewhere the source control server wasn't, things could get painful. I remember using a tool called Rational Clear Case that was setup in such a way that it took me most of the morning to commit a few files to source control if I was working from home. As bad as that sounds, it could be worse — if you were on a plane or somewhere without internet access, you wouldn't be able to work at all, unless you had planned ahead of time to acquire an "offline" version of the code. And then, getting it back online could be quite painful once you re-connected.

Git, the version control system upon which Github is based, changed all that with distributed version control. Git was the version control of Linux — a decentralized, democratic tool that could support ad-hoc, global collaborations. Github, the website, wrapped Git up and encouraged you to work with the full safety of source control wherever you were. And, don't worry, they reassured you, it'll be a breeze to sync back up when you're connected again. And it was.

GitHub offered remote coding to an increasingly remote workforce.

Social Coding

An early motto of GitHub was, "social coding." In today's world of ubiquitous technical meetups, conferences, hackathons, and generally extrovert-friendly activities, the idea of social coding seems sort of like a given. But if you were to go back 10 years and examine the anatomy of a user group, you'd discover a much more reserved affair, with a much different demographic. They'd be drinking Diet Coke instead of craft beer, and they'd have antiseptic discussions about industry trends.

Is it a coincidence that the show, "Big Bang Theory" arrived roughly at the same time as GitHub and that both rocketed to popularity? Yes, absolutely. It's a total coincidence. (If you thought I was going to say otherwise, gotcha!) But the coincidence itself offers not one, but two data points on the rise of "geek culture." As the demand for software developers (and their pay) have skyrocketed, "geek culture" has gone incredibly mainstream.

GitHub greased the skids for that. Nobody had thought (to my knowledge) to marry the concept of a social network with the concept of code, but they did. They encouraged you to share source code — to take pride in it — and to follow other developers. What's more, they brought the young, high-energy culture of Silicon Valley to the internet for broad consumption, and invited developers around the world to join in. Let's all come together, code, and be proud of who we are and what we do.

On the Shoulders of Giants

15 years ago, open source software was a radical idea, put forth by fringe participants, communities of hackers, and academics. Thinking back to early in my career, the idea of open-sourcing a side home automation project of mine was crazy. What if it started to take off?! I'd be giving away my precious, million dollar idea!

Nobody thinks that way anymore, outside of enterprises frozen in amber (whose code, ironically, no one would want). Your value is not in your code itself, but in your ability to write code, your branding, and your community presence. If you can point to a popular, open source, expert system that you've written, you can secure $200/hour consulting gigs. Angrily smacking someone looking over your shoulder at your code is no path to profit.

GitHub itself was at the core of this movement. You could realize the awesome power of distributed source control through a nice GUI and look cool doing it… as long as you accepted the premise that you had to share your toys (or pay a premium). The surprising value proposition of GitHub turned out to be so high that people gradually released their vice grip on their code.

Taking Over the World

GitHub, it turns out, offered the promise of distributed effectiveness, sociability and personal cred. Does this add up to zeitgeist — to taking over the world? To answer that, let's re-frame a bit.

Software developers demand the ability to work effectively from anywhere. They have attained a coolness factor, and demand for them is so high that there is no need for them to guard their source code like squirrels preparing for winter. GitHub is a good idea because it effectively captured what software developers really want and offered it to them pretty flawlessly. GitHub is a zeitgeist that is taking over the world precisely because software developers are taking over the world and software developers really like GitHub.

Related Refcard: Git Patterns and Anti-Patterns

The DevOps Zone is brought to you in partnership with New Relic. Read about where DevOps started and how the adoption rate went from tiny startups to giant Fortune 500 enterprises.