

# It's All about Collaboration

## Open source and the future of software

By Monica Pratt, *ArcUser* Editor

"Software development isn't really about software; it's about people," said Chris Wanstrath, cofounder and CEO of GitHub, in his Keynote Address to the attendees of the 2014 Esri International Developer Summit (DevSummit), held March 10–13 in Palm Springs, California.

During the hour-long address, Wanstrath described his journey over the last nine years from Cincinnati to San Francisco and from a disenchanted English major to cofounder of the largest social coding site. During the course of this journey, he changed from a fervent believer in open source who secretly—and with some pangs of guilt—dreamed of running his own business to the pragmatic head of a hugely successful corporation who sees open- and closed-source software development as complementary and, ideally, intertwined.

### A (Very) Brief History of Wanstrath

As he energetically roamed the stage, Wanstrath seemed to be having an animated conversation with friends rather than giving a speech to a hall packed with nearly 1,500 developers.

He described himself as a builder who "always liked to draw; liked to write; liked to make music; liked to build castles of mud, sand, stone...any raw materials I could get my hands on." When he was about 12 years old, he was messing around with an ancient computer in his grandparents' basement when he stumbled on a copy of QBasic. He discovered that building software was "really magical," and he was hooked.

His focus moved from writing QBasic programs to video game programming to programming for the web and trying to understand more about human-computer interaction. Although he programmed a lot in high school, he decided not to go the computer science route in college. Instead, he majored in English.

He also became active in the open-source community. "I really latched onto the philosophy of open source: that information wants to be free (except for my bank account PIN) and that sharing code is the moral way to do it. Proprietary code is evil—all that sort of crazy stuff you get into when you are very young." At the same time, he longed to start his own business because he believed strongly in "doing things right and really being proud of what you are doing."

In 2005, he realized he was spending all his time and money programming, so he dropped out of college, moved to San Francisco, and got a job as a programmer at the video game site Gamespot. While he liked the user community, his job largely involved pleasing advertisers. He quit Gamespot in 2007 and spent the summer consulting.

During this time, he became aware of Git, an open-source version control system that had been invented in 2005 by Linus Torvalds. Famous as the creator of the Linux operating system,

Torvalds had created Git as the way to manage work on the Linux kernel. It was Git that sparked the invention of GitHub.

About the same time he started working with Tom Preston-Werner on what would become GitHub, he was also working on a project called FamSpam, an app designed to solve problems with intrafamily communications. It turned out to be an idea that everyone liked but a product that no one wanted. In retrospect, Wanstrath realized that he had spent too much time on the idea and never had the right problem. "Technology should make your life easier, not harder." Instead of inventing FamSpam, he "could have just picked up the phone and talked to my mom, but that just wasn't happening at the time."

Both projects were released to beta at about the same time, but Wanstrath quickly found himself spending all his time with GitHub.

### Why and How GitHub Works

The vision for GitHub was initially very limited: code sharing without aggravation. Preston-Werner and Wanstrath reasoned that the worst thing that could come out of the project would be an awesome tool that no one else would use. So what? They would still have an awesome tool and have a great time using it.

Wanstrath believes a large part of GitHub's success is due to its flexible model that does not require permission for collaboration. Any project can be forked (copied down), changes made to the copy, and—optionally—a pull request can be made to the owner of the original repository who can decide to merge the changes (or not).



Originally, forking involved taking a philosophical stance against someone's code, but GitHub has made forking a loving term. "The real meaning of fork you: Let's build something together and build something great," said Wanstrath. GitHub continually refines the product by finding ways to make collaboration and communication easier by eliminating friction from the process.

The company's philosophy is that the product *is* the marketing, and the way to accelerate growth is to make the product "super good." The response to GitHub has been overwhelmingly positive. In just five years, it has become the largest code hosting site in the world with more than 5 million members sharing more than 10 million repositories. GitHub's home page says "Build software better together," and for Wanstrath, that is what it is all about—not git, not code, but projects and people working together.

People can use GitHub to work open-source code publicly via the web or work privately on closed-source code in a private repository on private servers using GitHub Enterprise, as Esri and other large software shops do.

## Not One or the Other

GitHub is a closed-source application built both on and with open-source software. "Our business would not exist if not for open source and closed source. It's the combination that makes GitHub successful. GitHub chooses to use open source because we think it makes sense," said Wanstrath. "We don't have a huge philosophy around it other than we like things that work well and we like things that are high quality. Our experience so far has been that open source is all those things."

Open source is "a really cool way to build software." GitHub uses GitHub to develop GitHub. Some software companies call using their own tools *dogfooding*, a term that he finds objectionable. He prefers the phrase "drinking our own champagne" because GitHub is a really cool tool that they would have used anyway.

Why do people do open source? There are many answers to that question. In some ways, open source just makes sense. It is less work to modify code and share it. And surprisingly, "when you share code with the world, the world will often give back." By sharing things that aren't core to its business, a company can get enhancements that benefit it and the larger community.

Why contribute back work? It can actually be more work not to contribute code back once patches, bug fixes, and other changes have been made because maintaining that code separately will be a lot more work.

However, some aspects of open source don't seem to make sense, at least on the surface. "People are just submitting a bug because it's simple. There's not something master planned behind it." That can be baffling for people who are looking for an angle, for the "what's in it for me" aspect of open source. They often overlook the simple fact that programming (like guitar playing and other activities) can be both a profession and a hobby.

"For a lot of people, this open-source thing is not a big mystery. They are just doing it for fun. There are people who work on the Linux kernel who are not professional programmers at all," said Wanstrath. "They're like a neurobiologist or something and they program on the Linux kernel for fun. That is what they do in their spare time. The world is a very strange place."

The impact of open source on business will only increase. Wanstrath believes the value of open source is not about the actual software but "the things you do with it and the way you think about building software. People who think open source is not right for them are as shortsighted as people who think everything should be closed source. The future is going to be this gray mix of both, and we are already there—it's just not evenly distributed."

## Harmonious Relationship

Wanstrath views the relationship between open source and closed source as a symbiotic, rather than adversarial, relationship. "There is not this big divide. It's not open source versus business. When I was young, that was what I thought. I thought business was evil for some reason because the code was kept secret and the code wanted to be free. That's not true at all. If anything, they complement each other."

Facebook, Esri, and many other companies build software both on open source and closed source and, in the process, release their own open source. The Esri site on GitHub is an example of the way things are going: a closed platform with open source on top of it. Currently Esri has hundreds of open-source projects on GitHub, including the projects posted as part of the JavaScript coding contest, 100-lines-or-less-js, held during the DevSummit.

The pattern of using both open and closed source is both popular and successful because it lets companies learn about the people using their software. Working directly with people who are building on a company's software—directly in the code—is a great way to do that.

Open source is all about community. Wanstrath maintains the lone coder is a myth. Software development is now a group activity. "Great coders relentlessly iterate. And I think the best coders relentlessly iterate with other people. They build things together. The future of this stuff is about the way people work together." Ultimately, successful software development is about getting stuff done and shipping software.

## Open Source for More than Software

A lot of times, open source is not about having a great idea up front but about releasing an idea and seeing how things develop. This approach has proved fruitful for more than just software—lots of activities can benefit from community and better communication. GitHub's own lawyers use GitHub to draft documents. The City of San Francisco posts laws on GitHub. The Smithsonian has posted the sunflower genome. People have written books using GitHub.

Wanstrath sees his potential customers as "anyone who builds stuff, anyone who wants to build stuff." GitHub is just a tool. Its power resides outside the tool in a workflow that everyone can participate in and understand.

"If I had to think about what's the future of programming (because I'm up here and it's like a keynote), it's not about fancier features, not about fancier testing, not about fancier code coverage—it's about fancier communication." Success will continue to come not from adding features but from removing them to create a workflow that makes it really easy to work with people. "With GitHub, you can work with anyone on anything in the world. It's all available to you," he said.