The GitHub Effect

David Alan Grier, George Washington University

Although GitHub has become a great tool for studying the software development process, it poses a fundamental problem to those who study work.

"Nice code": unremarkable words linked to an unremarkable program buried deep in the open source software development site GitHub. I had been trying to get a better understanding of the GitHub community by reviewing project repositories and logs. It wasn't proving to be a very profitable experience, as it seemed to reveal a group of programmers who were creating software in fairly conventional ways. For some reason, the comment caught my attention—perhaps it could show me something that I was missing.

GitHub has become the community center for the current generation of software developers. More and more of my active developer friends have embraced the site over the last five years. Not long ago, they would have found support and camaraderie in a university computer lab or a collaborative workspace. Now, they log into GitHub and follow a stream that talks of commits and pulls, forks and merges. They get all the support they need from a distributed community that's linked by a distributed work site.

As I studied the site, I concluded that GitHub had much in common with social media—or perhaps social media borrowed ideas from software developers. The interaction on GitHub was vigorous and filled with slang, emoticons, and irony. The phrase "nice code" could have been a note of reassurance and support, or it could have been a sarcastic slap in the face—the Internet way of saying "poor work." At best, I could tell that it was connected to a change that altered neither the function nor the logic of a larger program.

GitHub has also become a great tool for studying the software development process, as the site collects and stores great amounts of data through comments and changes to code. For the most part, scholars have drawn only modest conclusions from their study of the GitHub database: they've found that source software development is mostly done by programmers employed by services companies, that these programmers come from relatively few major cities, and that programmers like to work in the evening but produce better code during the day. To no one's surprise, they also claim that a small number of programmers do a disproportionate amount of the coding.

Any type of work is fundamentally difficult to study. One of the more common problems is the Hawthorne effect—the tendency of workers to recognize that they're being studied and to change their habits to meet the expectations of the researchers. Psychologist and theorist Elton Mayo, who discovered this effect, argued that it's too often assumed that almost any young researcher can charge "into industry, armed with some rags and tatters of scientific method borrowed mainly from chemistry or physics, and proceed to make some interesting findings." Such an approach usually fails, he explained, because it "ignores completely the mutual complexity and facts of human association."

One could claim that the Hawthorne effect doesn't touch GitHub because the site is largely public: the GitHub repository is public unless the developer pays an extra fee for a private service. Even visits to the repository are observed. For example, the time I spent reading GitHub log files was noted by two developers. One asked if he could give me a demo of his project, and the other wanted to know if I could contribute to her code.

Yet, GitHub poses a fundamental problem to those who study work. Even if developers aren't adjusting their habits to impress researchers, they seem to be aware that software development is now a public activity. However, public activities can encourage anonymity as much as they encourage people to impress their superiors. In public, people often strive to blend with the crowd. They generally dress as others dress and behave as others behave. Colleagues greet each other with phrases like "have a nice day," words that communicate politeness but don't encourage further interaction. In such a world, "nice code" could be a compliment or an insult, or a meaningless greeting that one developer routinely offers to another.

David Alan Grier is a professor of international science and technology policy at George Washington University. Contact him at grier@gwu.edu.