

VK Newsletter

The silent majority

 [Blog](#) |  [Newsletter](#) |  [Founders Guide](#) |  [CTO Guide](#)

Vadim Kravcenko - *28th August 2022*



Silent Engineers	3
Silent Users	5
In defense of being vocal	7
My thoughts	8



The “silent majority” was used by President Richard Nixon during his presidency and his campaign. In this usage, it referred to those Americans who did not join in the large demonstrations [against the Vietnam War at the time](#), who did not join in the counterculture, and who did not participate in public discourse. Nixon, along with many others, saw this group of Middle Americans as being overshadowed in the media by the more vocal minority.

We're not talking about politics. This interesting concept of the majority being overshadowed by the vocal few is quite fascinating and holds true in software engineering.

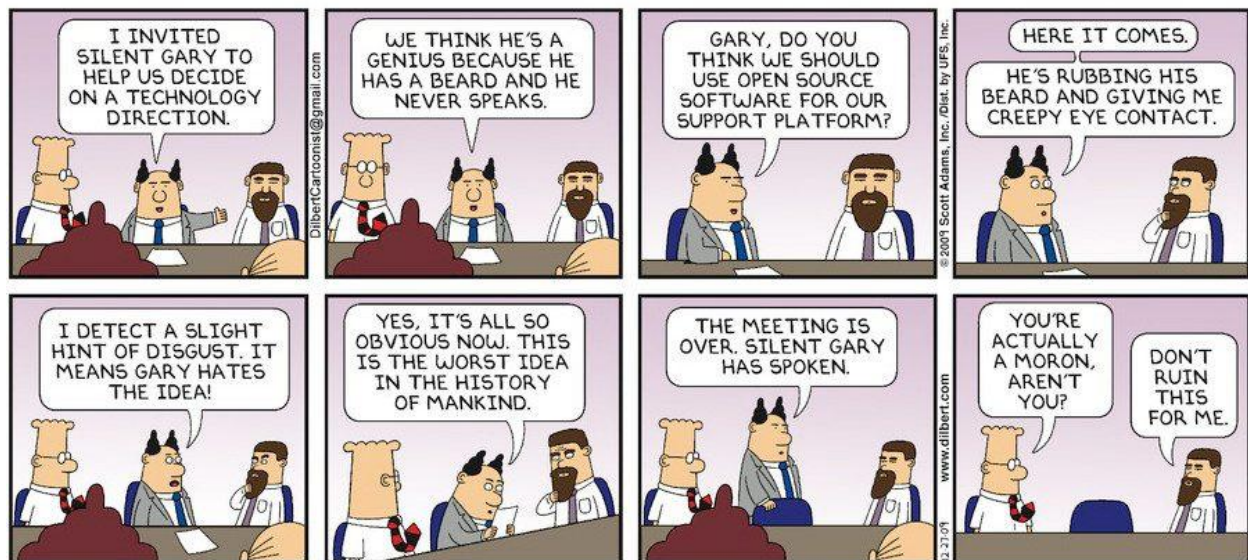
In software development, the silent majority are the engineers who write the code, debug the programs, and solve the complex issues behind the scenes. They do not participate in controversial discussions about Visual Basic or Pascal — they just do their work in those languages without even knowing that there's so much controversy surrounding their language of choice.

Without this silent majority, many projects would, in fact, grind to a halt. It is often their quiet diligence that keeps a project on track and prevents it from falling apart.

There also seems to be an assumption on HN/Reddit that vocal activity on the internet, in any form — be that videos, blogging, podcasts, etc. — is proportional to activity behind the screen. If you're constantly seeing stuff about crypto, then you're probably scrolling Twitter, but if you leave that bubble and go outside — most people don't care.

Silent Engineers

While browsing HackerNews, I sometimes get the feeling that every developer out there is working for FAANG, as there are always posts from those people doing some hyped stuff. Or you might think that PHP is never used nowadays because whenever it's mentioned, everyone is hating on it in the comments.



Dilbert and the silent engineer. Not really relevant, but still funny.

But let's be straight, that's like 1% of all of the developers out there — the rest of them are just lurking and coding with their language of choice and being content with it. Be it Fortran, COBOL, Perl, or PHP. I've seen so much hate some languages get that I'm surprised anyone still writes code in those languages, but then I



remember that [everything is subjective](#), and the articles that I read represent every small subset of developers.

Even HackerNews is not that popular — I know many great engineers who've never visited the website. There are so many articles and comments by people whose level of enthusiasm doesn't match their experience. Maybe also my own, but I just like writing, so deal with it.

Usually, the comments on HN/Reddit are polarised by a single group of people who have the same opinion, and then it's hard to object and present a different perspective, even if you speak with more experience and context than the masses.

It's also important to understand that we have a generational divide among software engineers. There are thousands of new software developers each year that have been taught differently than the previous generation. This introduces a bias to the particular expertise that gets shared.

Some developers signed so many NDAs over the years it almost looks like they did nothing at all.

I really like that some subset of the silent majority still participates on GitHub with bug fixes to their favorite libraries. Sometimes I've seen Pull Requests from empty accounts with a brief explanation of what was implemented. They just submit bug fixes, no drama.

Silent Users

I'm sure you're aware of the importance of customer feedback. After all, it's essential to know what users think of your product to improve it. However, there are users who never give feedback, either because they're happy with the product as it is or because they don't bother to take the time to fill out surveys and submit bug reports — the silent majority of your customers.

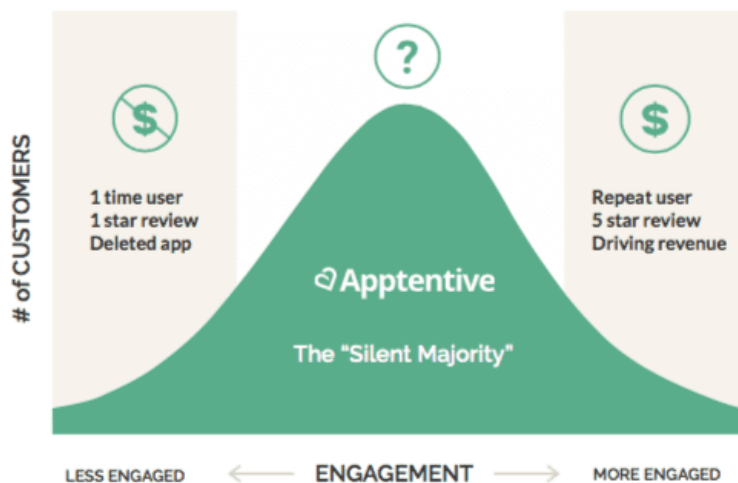


Even Jerry, with his years of requirements elicitation experience, couldn't figure out how to get the requirements out of these customers."

Dealing with your silent customers is hard

As a result, companies often have a skewed view of their user base and improve the wrong things, thinking that the only people they should optimize for, are the people who fill out their “what did you like about this service” surveys. I never fill out those btw, it’s a waste of time. If I’m using a service, I’m already satisfied with it. Otherwise, I would jump to another one.

You can't rely on silent customers to give you honest feedback, but you can still learn a lot from them. Observing how they use your product is the first thing and setting up proper analytics to get an insight into their needs and expectations is the second.

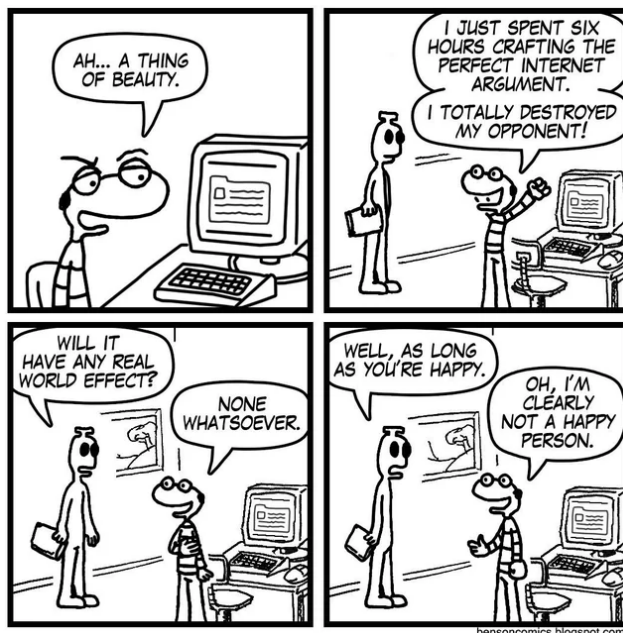


The problem with silent customers is that while they often demand very little, they will also silently switch providers if they're not happy.

In defense of being vocal

Being vocal is hard. It might seem easy — you just write an article or make a video — but there's a reason why only a small percentage do it. It takes huge amounts of time. Even this small newsletter issue took me a few hours on my weekend to write. Not everyone is willing to do the work just to bring their opinions to the masses.

It also takes confidence — whenever you voice an opinion on the internet, there will always be people who have an opposite one, so you need to prepare yourself to read tens of comments that disagree with you. Reading negative comments can be disheartening, but it's important to remember that not everyone will agree with you. And that's fine, we're all amateurs, and sometimes we can be wrong.



Sometimes people write comments just to argue



My thoughts

And now to my final thoughts. When it comes to the software community, there are two schools of thought. Some people believe that it's important to be vocal and share your opinions, while others believe that it's better to stay quiet and let the quality of your work speak for itself. Personally, I believe that being more vocal can only be a good thing.

First of all, when you're vocal, you're more likely to be heard. If you have something valuable to say, then you owe it to yourself and to the community to speak up. Secondly, being more vocal can help to create a more inclusive community. Too often, online conversations are dominated by a small subset of people. By speaking up, we can help to ensure that everyone's voices are heard.

Of course, you can get downvoted, but who cares?

In many cases, fear is what's holding us back - fear of criticism, or of saying something stupid. But if we want the software community to thrive, we need to get over that fear and start speaking up. It's time for us to be bolder and more vocal. Only then can we hope to create a truly inclusive community where everyone feels welcome and valued.

