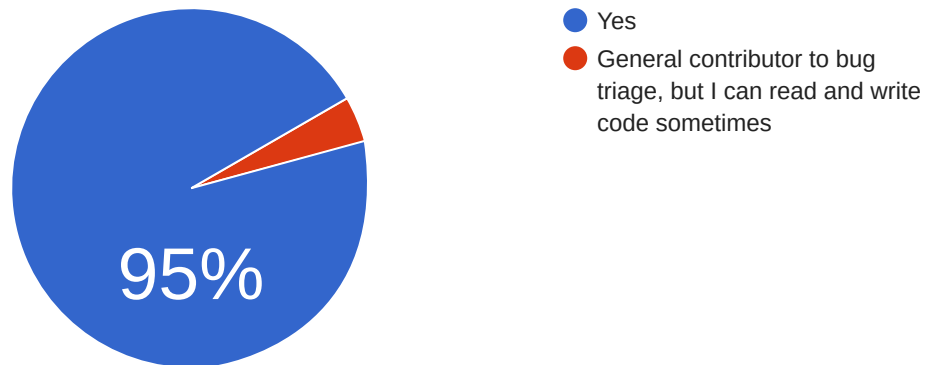


Inkscape Developer Survey (2024)

24 responses

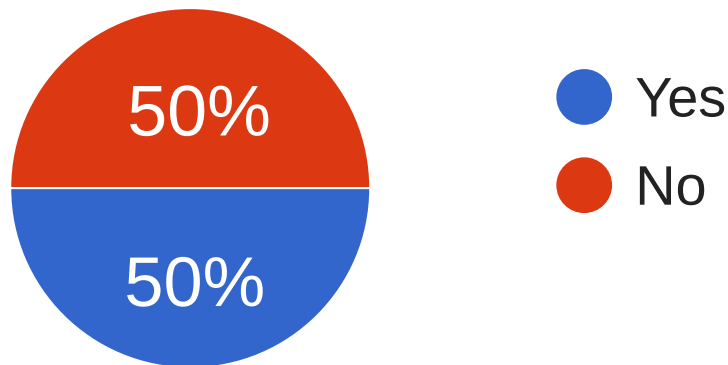
Are you a programmer?

24 responses



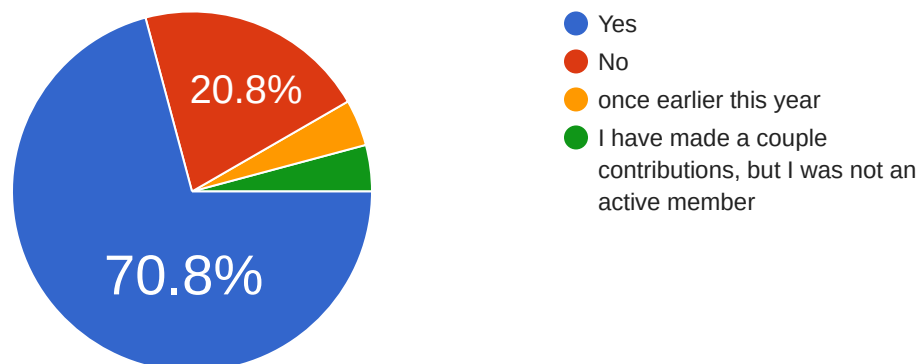
Do you consider yourself currently active (2024) in the Inkscape Project?

24 responses



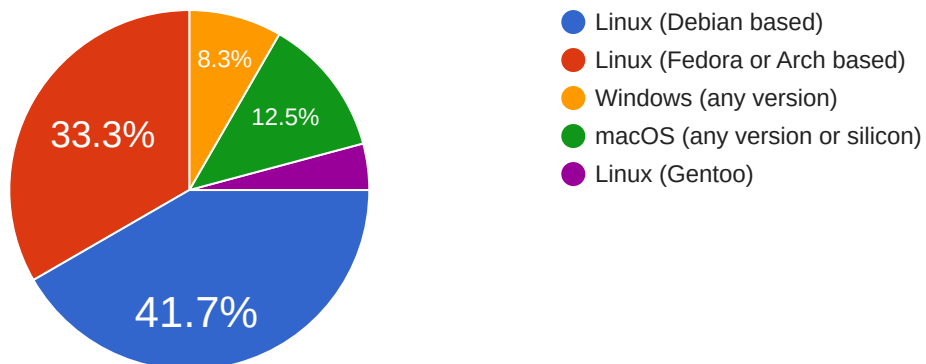
Were you actively contributing in previous years?

24 responses



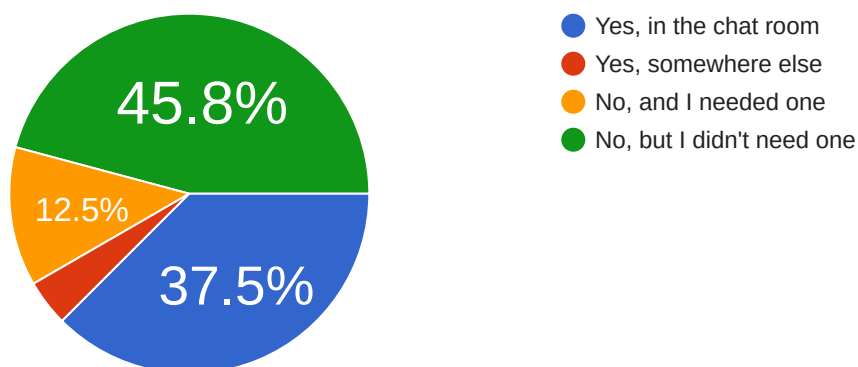
What Operating System do you use to program?

24 responses



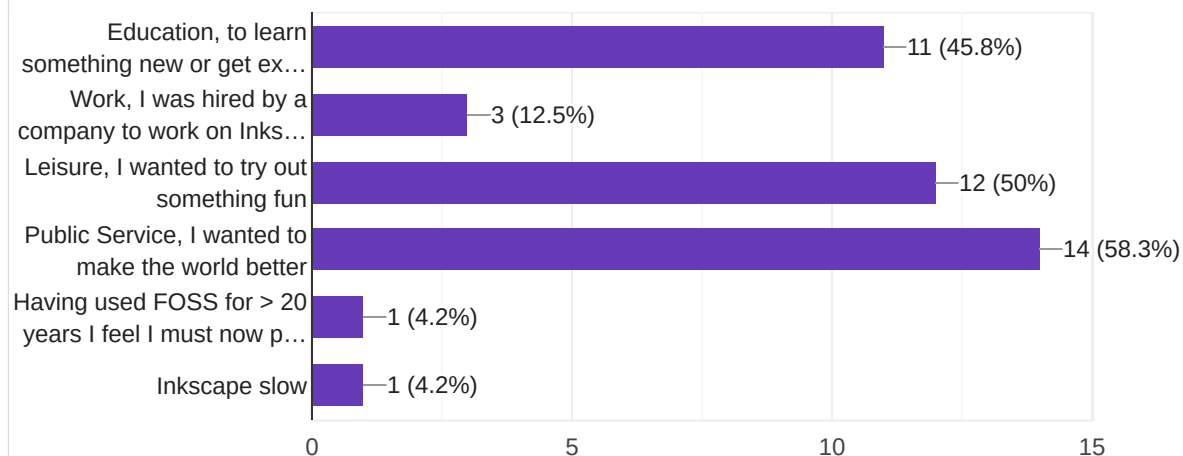
Did you find a mentor to help you get started?

24 responses



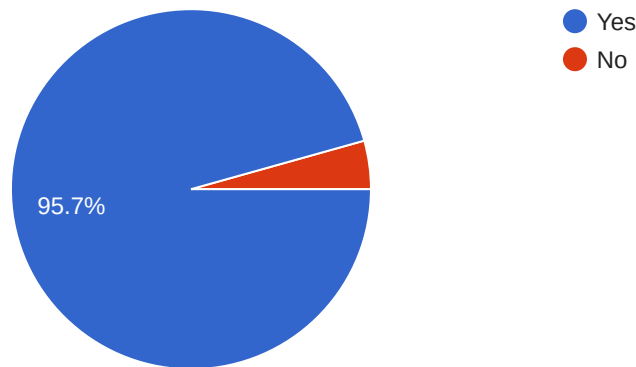
Why did you want to work on Inkscape?

24 responses



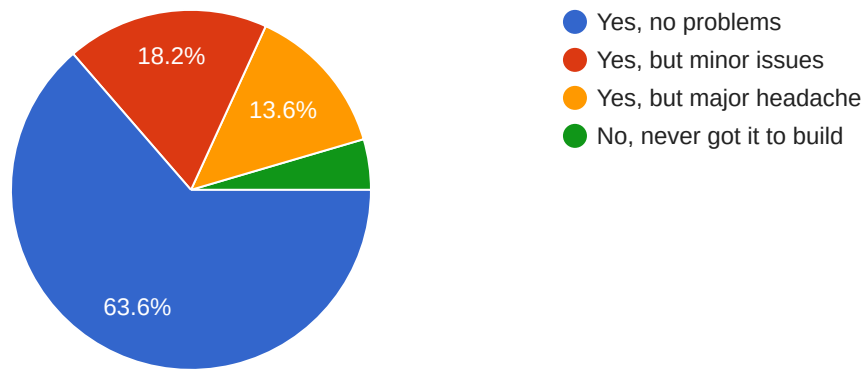
Did you find the build instructions for your operating system?

23 responses



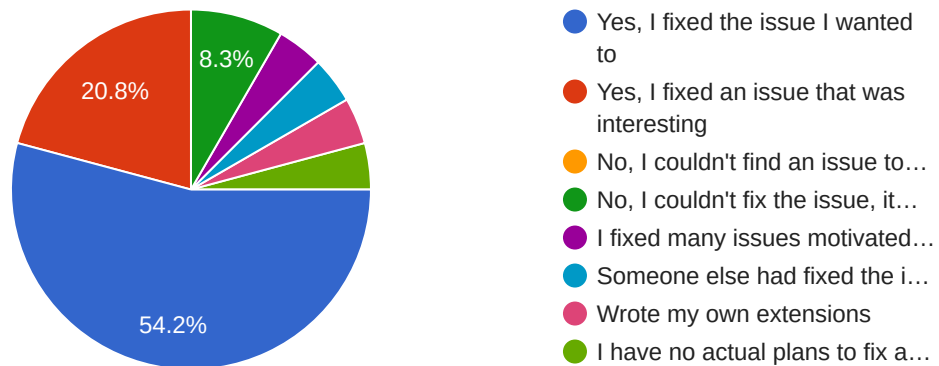
Did those instructions work?

22 responses



Did you fix an issue for Inkscape?

24 responses



What would you say Inkscape needs to work on to help programmers continue contributing to the project in the long term?

20 responses

Better collaboration tools (chat)

Less scattering of instructions

~_(\u2197)_/~

Filling out a quiz

I don't want to guess; we need to ask one-time contributors.

Make contributing as easy as possible. Language (spoken and programming), GIT, raise the bar already. Clear build instructions help. Maybe Windows build system that doesn't suck?

Build instructions for Inkscape MacOS are not very reliable. Would be great if someone could take a look whenever they have some spare time. Thanks!

I am not certain yet. I think a good part of it might be my skill with C++ right now. The codebase has a very clean structure to an extent obvious even to me as a newcomer. Maybe a guide for what skills to develop in what order to become capable for inkscape.

Better build guide for gtk4 on MacOS

Review PRs soon

Updated step by step guide with all information centralized on it

A better guide on various libraries used.

Keep up the helpful and friendly community. What we lack is support from testers, bug managers, and a QA-like process (with clear guidelines). Things got worse after the departure of Nathan, who did a great job of coordinating the work of developers, suggesting issues to work on to whomever might have expertise on the subject, ushering in newcomers, managing the trackers, etc. Maybe the guiding of newcomers could be an assigned role in the project and can be separated from bug management, as it would be hard for a single contributor to fill in for Nathan.

When a Pull-Request is 99% finished and in the review phase, it feels like the responsibility is not clear. Even after a positive review, Pull-Request do not get picked up and merged fast in my experience. Instead, I had to actively push for someone. It would be much more engaging if pull requests get merged fast and if the developer knows if he is in charge to merge or when

not, that someone from the core team gets assigned.

Change the gui code language/framework such that it's easier to edit and make improvements.

Keep MR review times low; Automate some of the bug triaging work (AI based suggestions? Auto-response bot? Auto-close stale issues that fulfill certain criteria? etc.)

I recall that we have the dev wiki site which seems to be a great resource, but links to it dont seem to be prominent enough maybe. Having say a large button on the Contribute pages on the main site could help. Plus including links to the relevant wiki pages when newer developers ask questions can increase awareness of it perhaps (I'm not sure if that's the current culture or not).

Better documentation for extensions, including examples

I strongly encourage the use of almost verbosity comments inside the source code. Assuming that "the code speaks itself" means that only difficult issues will be fixed by experienced programmers.

Some high-level documentation about the architecture, DOM, the main classes, etc. would have been helpful for me in the past. For an occasional dev like me, its easy to fix a bug within some limited scope, but getting the bigger picture is more difficult

Prompt review/testing/merging of merge requests, especially for first-time contributors, but not limited to them. For Windows developers, a native compiling and debugging experience: MSVC not MinGW.

Any other remarks?

11 responses

It surprises me how few programmers Inkscape was able to attract. Maybe C++ is uncool.

Sorry for being inactive for the past couple of weeks, I've started working full time and have been going through some mandatory training at work. I'll be back as soon as I figure out a healthy work-life balance. In the meantime, I'll try to help with MR reviews and basic issue ;)

The "Operating System" question should be multi-select.

Thank you very much for the great job, I'm impressed by the work

Many good architecture-level ideas and design improvement plans were discussed during the 2024 Inkscape Summit. For instance, Marc proposed a new way to implement LPEs which would solve many issues with them. I would very much like for such ideas to be documented on the project Wiki so that the efforts on design improvements are better planned and coordinated. PBS made many improvements by refactoring the code, but these were often "local" improvements that introduced many issues at the higher, architectural level (fragile approaches to concurrent access and object lifetime, relying on programmers to do the "right thing" rather than solutions robust by design). I believe that such issues could have been avoided if design documentation had been created, shared, and discussed at more length. MR reviews should ensure that enough doxygen comments are added to describe newly added functionality at a lower level too.

The community is nice and fun to work with

Fantastic job! Amazing product from a small team of devs. The chat was just what I needed to help break into the code base. Ultimately, the code base was just a little too complicated to make the additional change I wanted to make, and the feature I was working on already had an alternate implementation done in parallel which took away the motivation.

I see two points where we create unnecessary friction: 1) It is annoying and unusual that we have a separate "inbox" project in Gitlab. This breaks some of GitLab's usability. Everybody else handles this by using tags and maybe some kind of bot. 2) The "getting started with development" page should be moved to somewhere where it can be edited easily (MD file in GitLab source repo?), and not require thousand translations (source code and bug reports are in English anyway, so what's the point of providing French compile instructions?).

Thanks for putting out this survey. I have been thinking on returning to an active member on this project.

I consider Inkscape as a foreign family and even if I'm not active as before, I try to keep myself updated with what's going on so I could have a great comeback.

I don't think anything is wrong with the community, and I don't see anything specifically challenging, other than my limited time. I'd love to spend more time on Inkscape, someday!