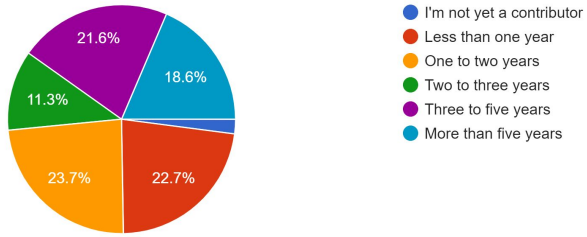
2020 Node.js Contributors Survey Results

Summary

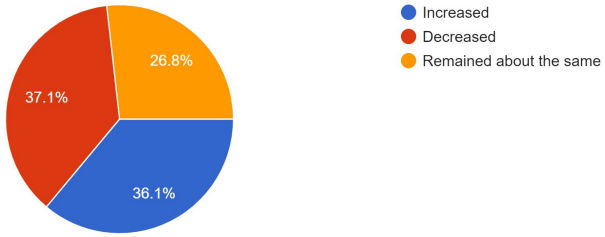
How long have you been contributing to Node.js?

97 responses



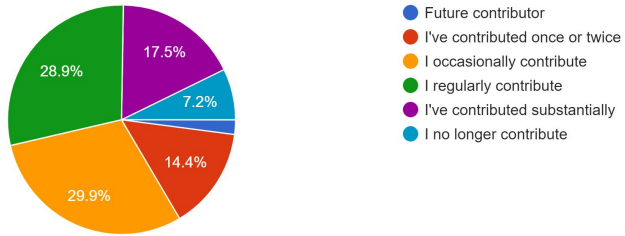
Over the past year, do you consider your contributions to have:

97 responses



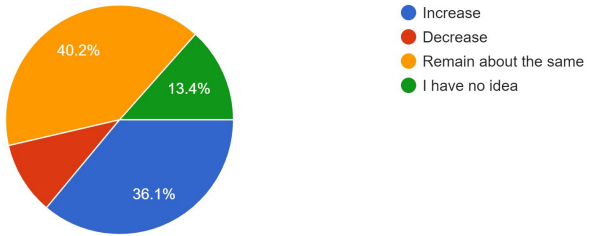
What kind of contributor do you consider yourself to be?

97 responses



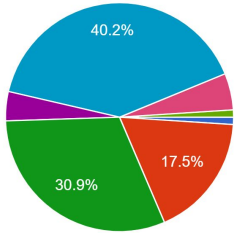
In the coming year, do you expect your contributions to:

97 responses

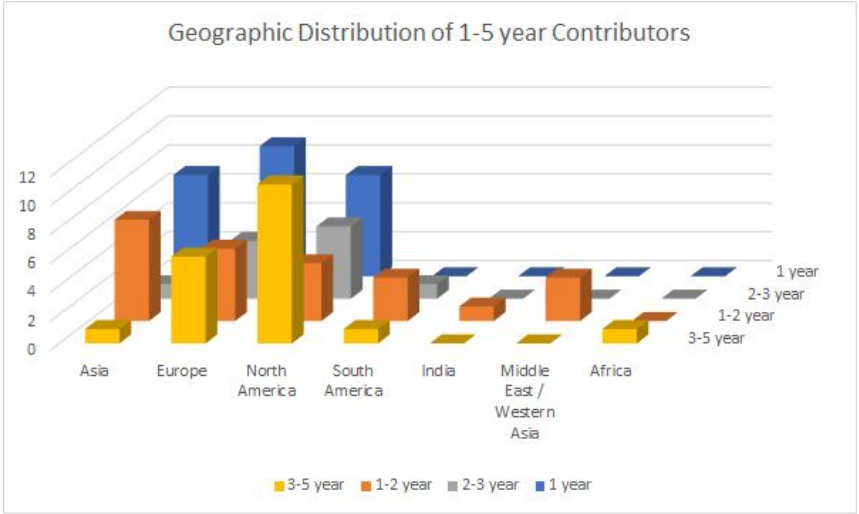


What region of the world do you live in?

97 responses

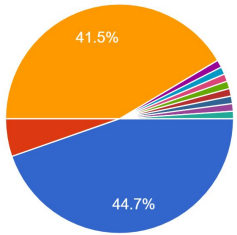


- Africa
- Asia
- Australia
- Europe
- Middle East / Western Asia
- North America
- South America
- India



Does your employer support your contributions to Node.js?

94 responses

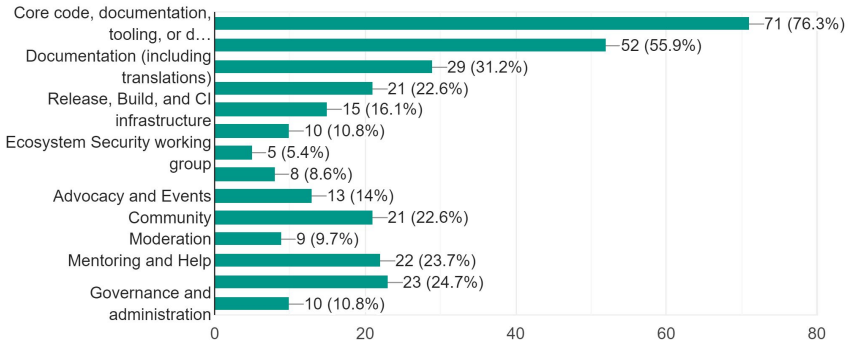


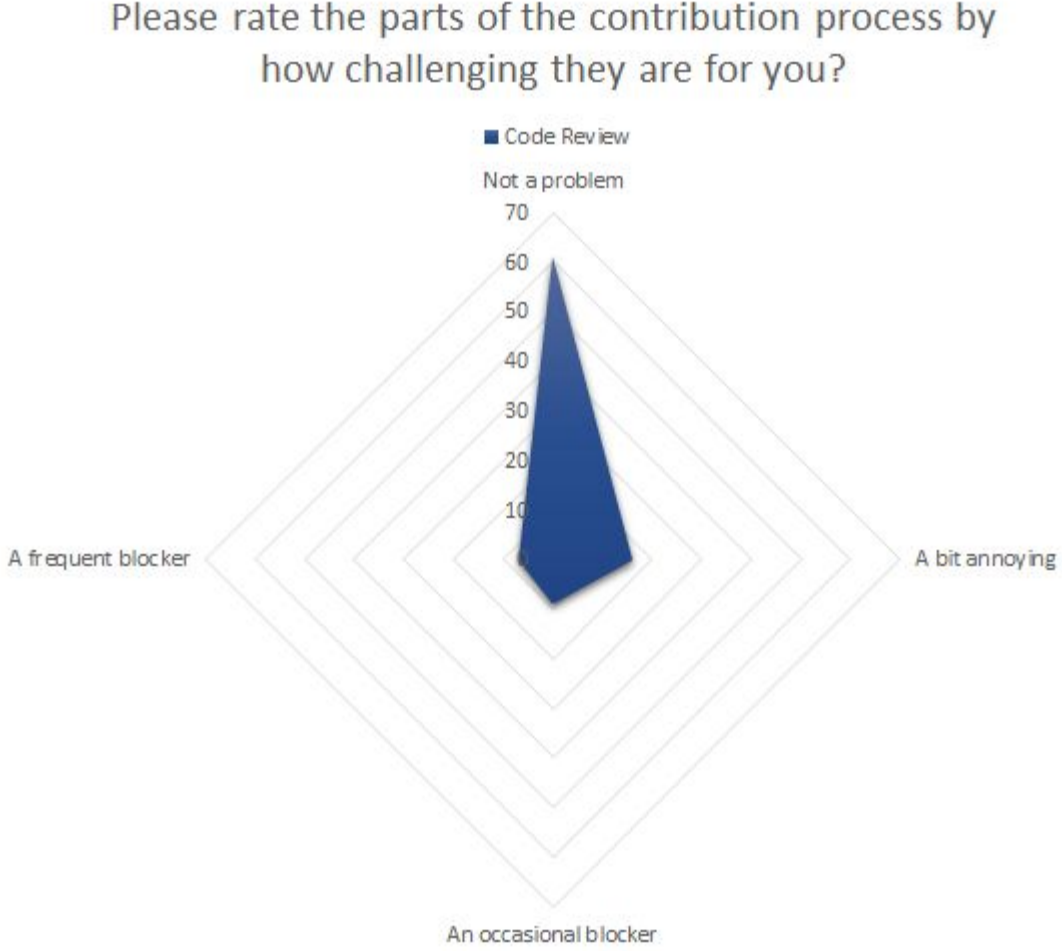
- Yes, I can contribute on company time
- Yes, I only contribute on company time
- No, I need to use my own time
- Not yet a contributor but my company…
- Just a student yet :)
- self-employed / can support OSS wor…
- I work for myself. So, yes and no.
- I'm currently on a three month contrac…

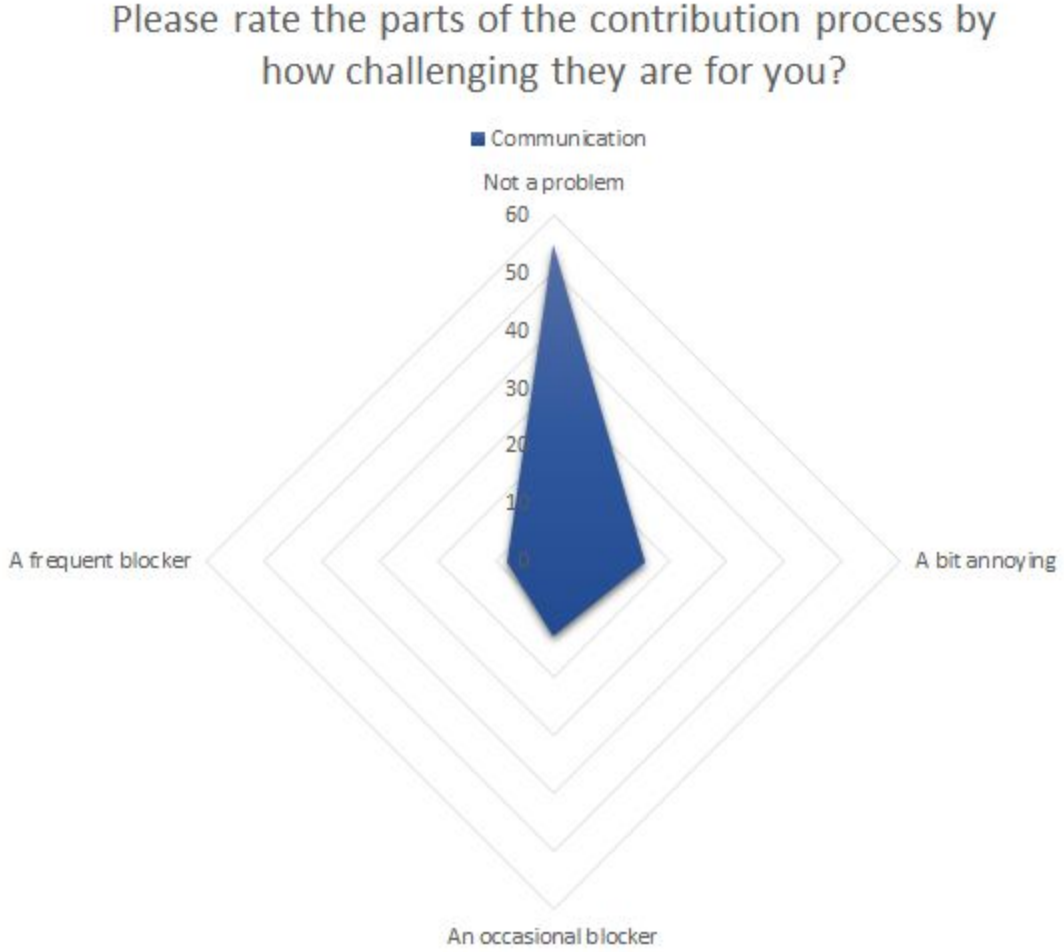
1/2

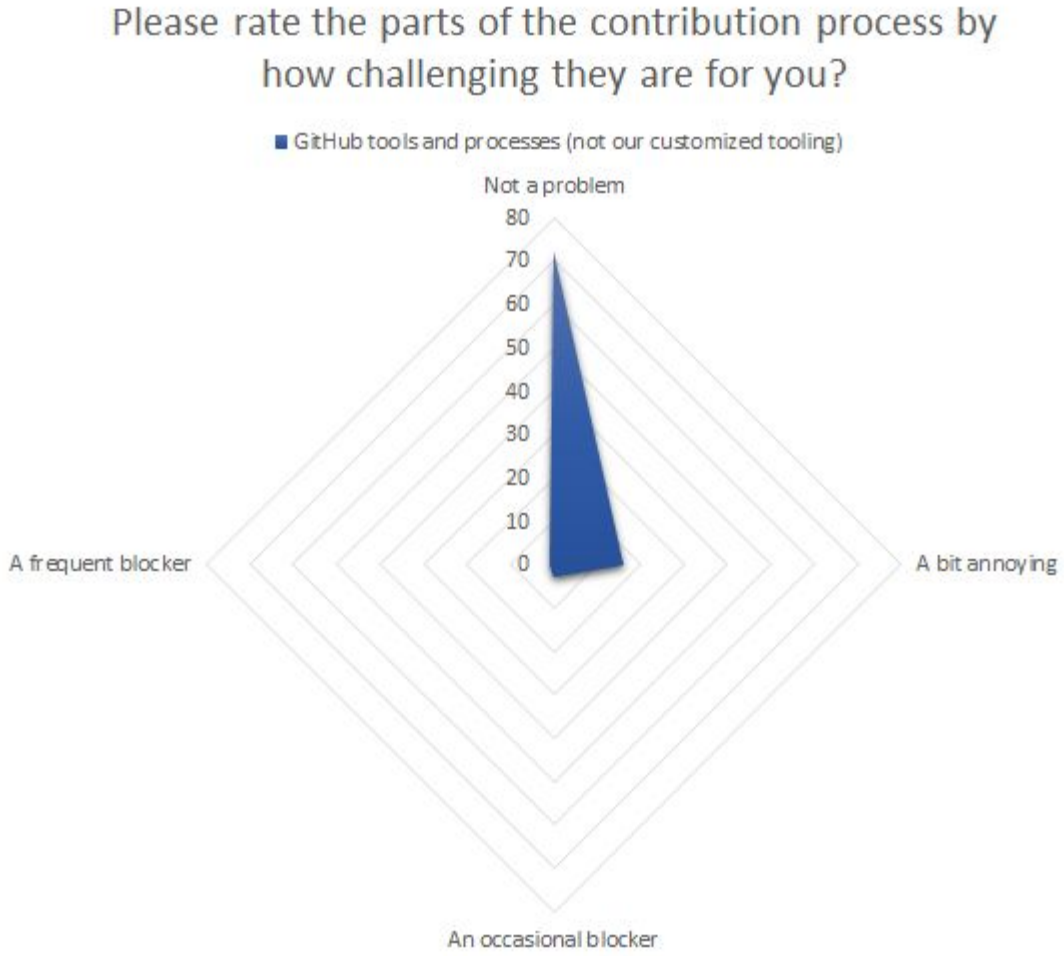
What areas of Node.js do you contribute to? Please check all that apply.

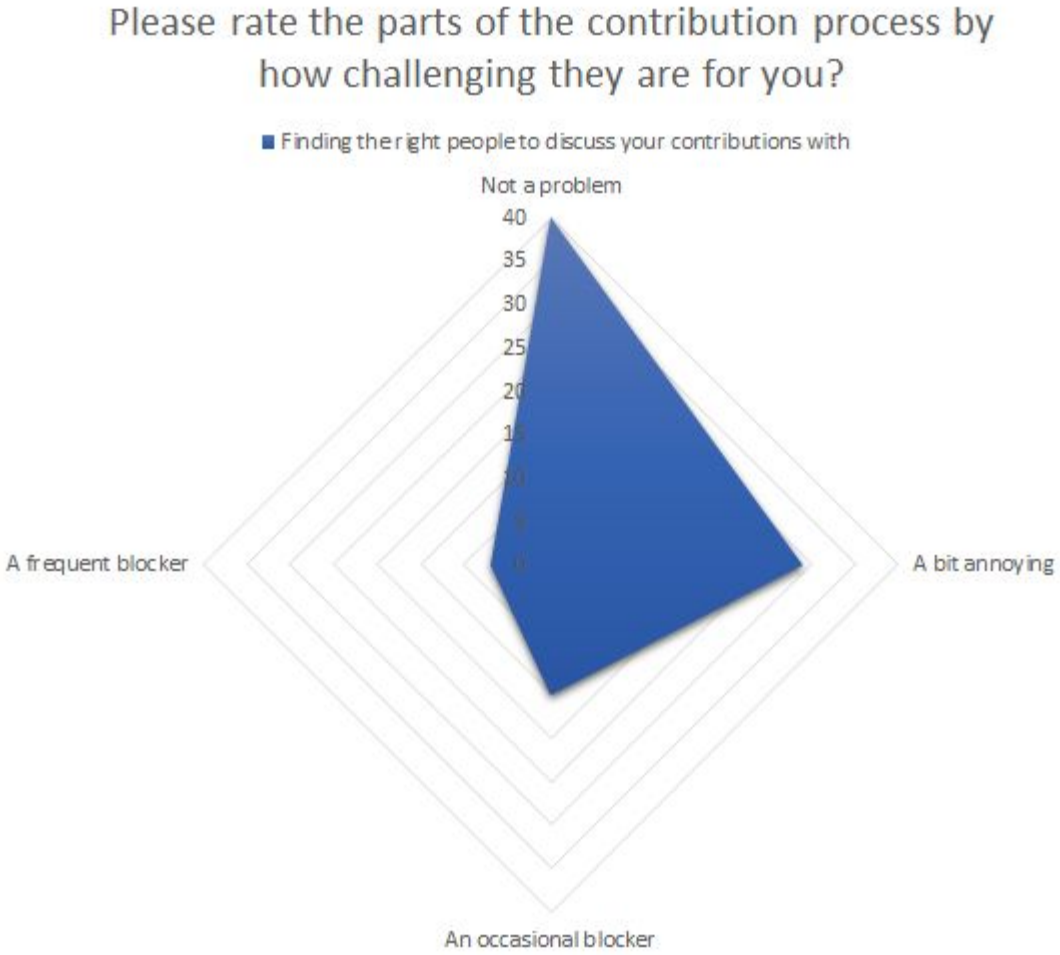
93 responses

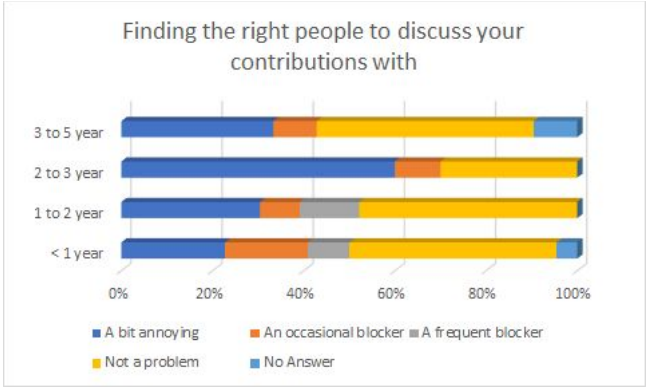


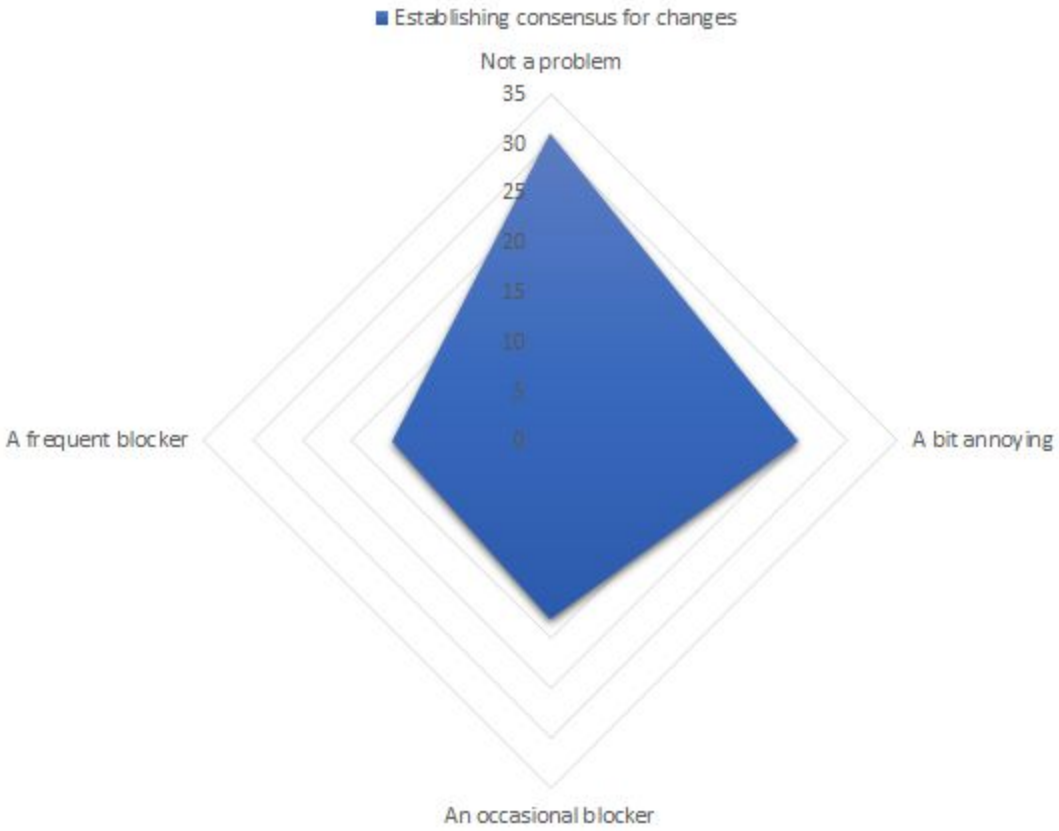

Please rate the parts of the contribution process by how challenging they are for you?
Code Review
Not a problem
70
60
50
40
30
20
10
A bit annoying
An occasional blocker
A frequent blocker

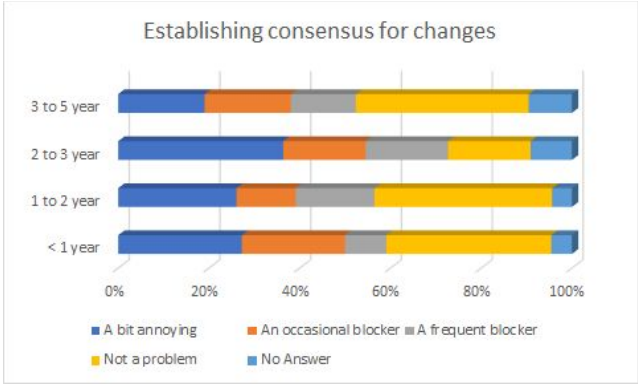

Please rate the parts of the contribution process by how challenging they are for you?
Communication
Not a problem
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker

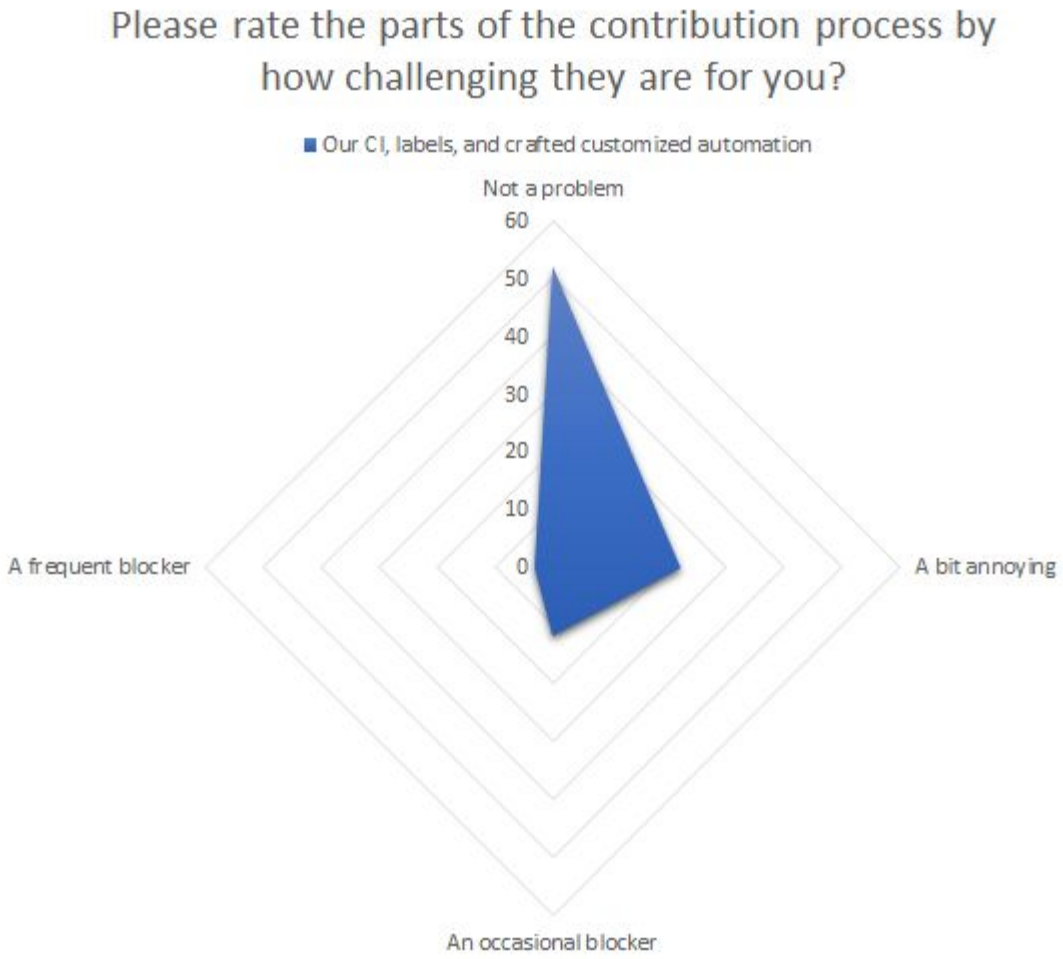

Please rate the parts of the contribution process by how challenging they are for you?
GitHub tools and processes (not our customized tooling)
Not a problem
80
70
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker

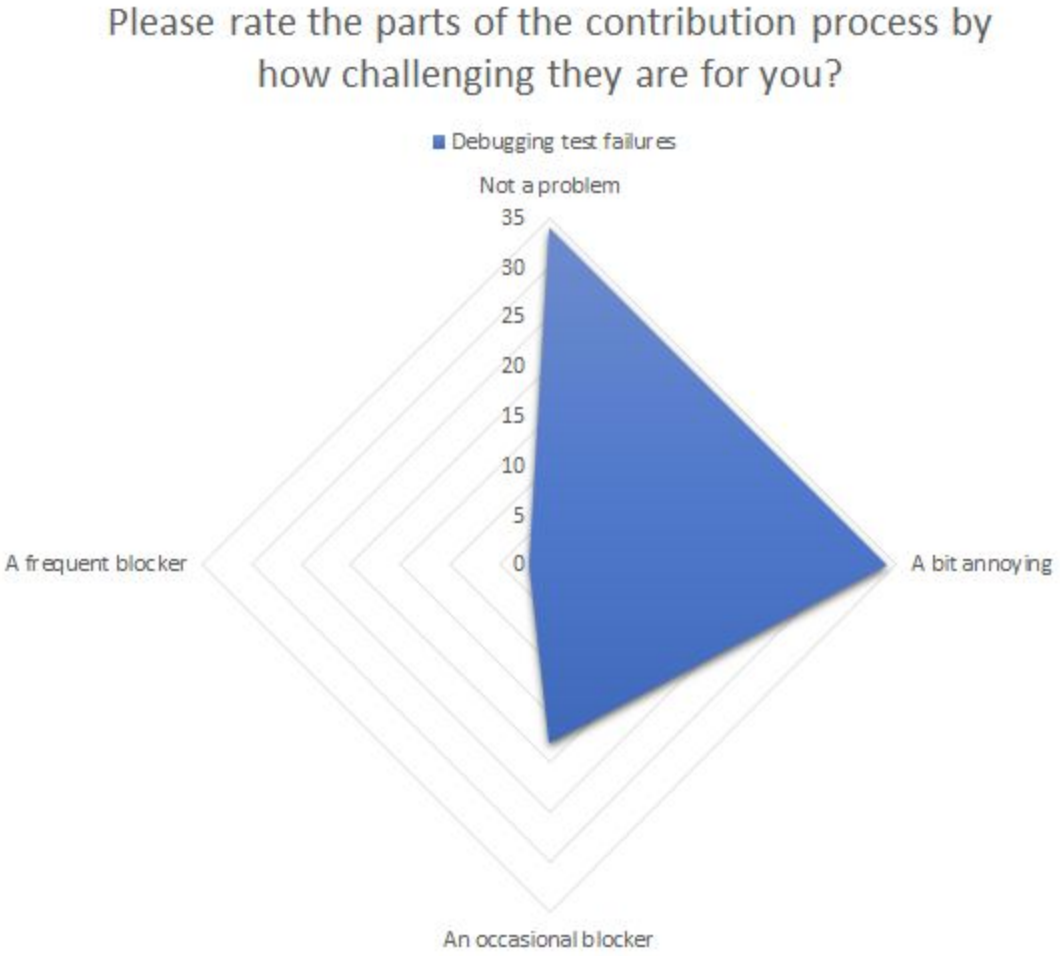

Please rate the parts of the contribution process by how challenging they are for you?
Finding the right people to discuss your contributions with
Not a problem
40
35
30
25
A bit annoying
An occasional blocker
A frequent blocker

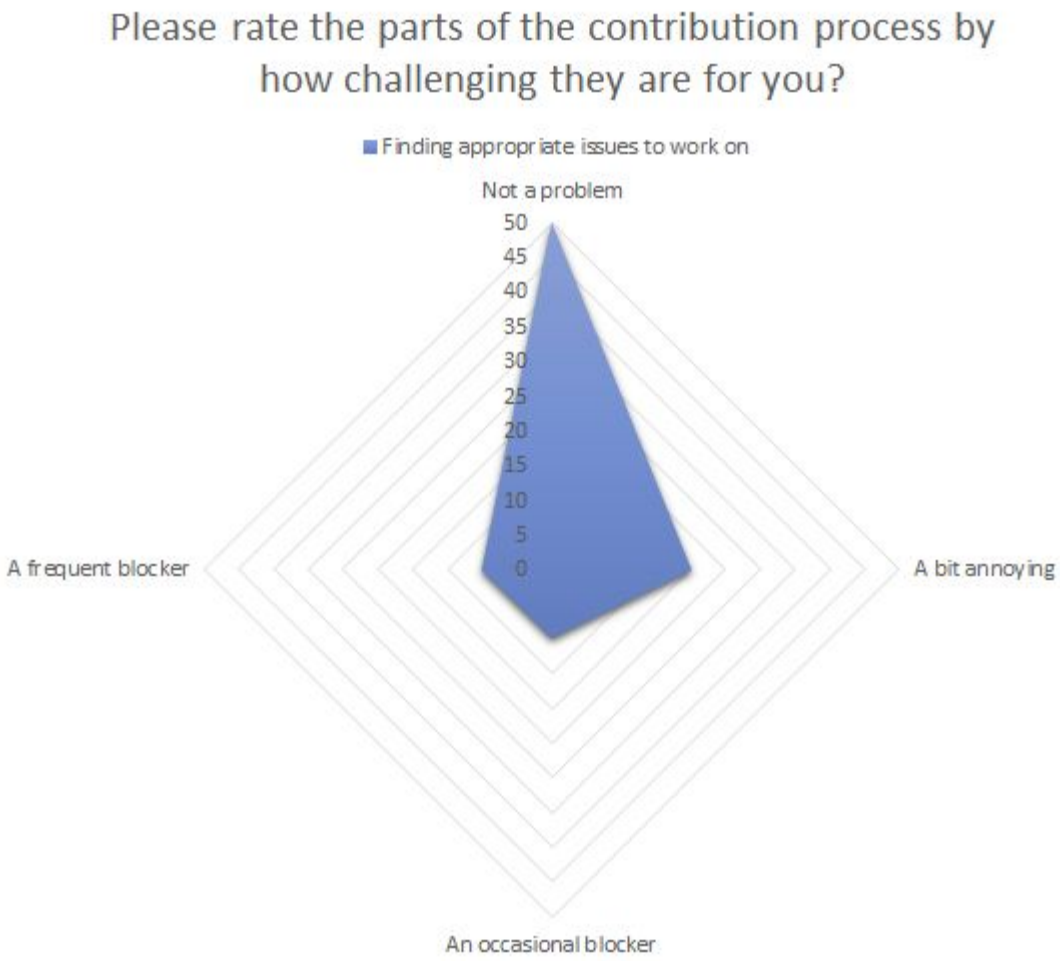

Finding the right people to discuss your contributions with
3 to 5 year
2 to 3 year
1 to 2 year
< 1 year
0%
20%
40%
60%
80%
100%
A bit annoying
An occasional blocker
A frequent blocker
Not a problem
No Answer

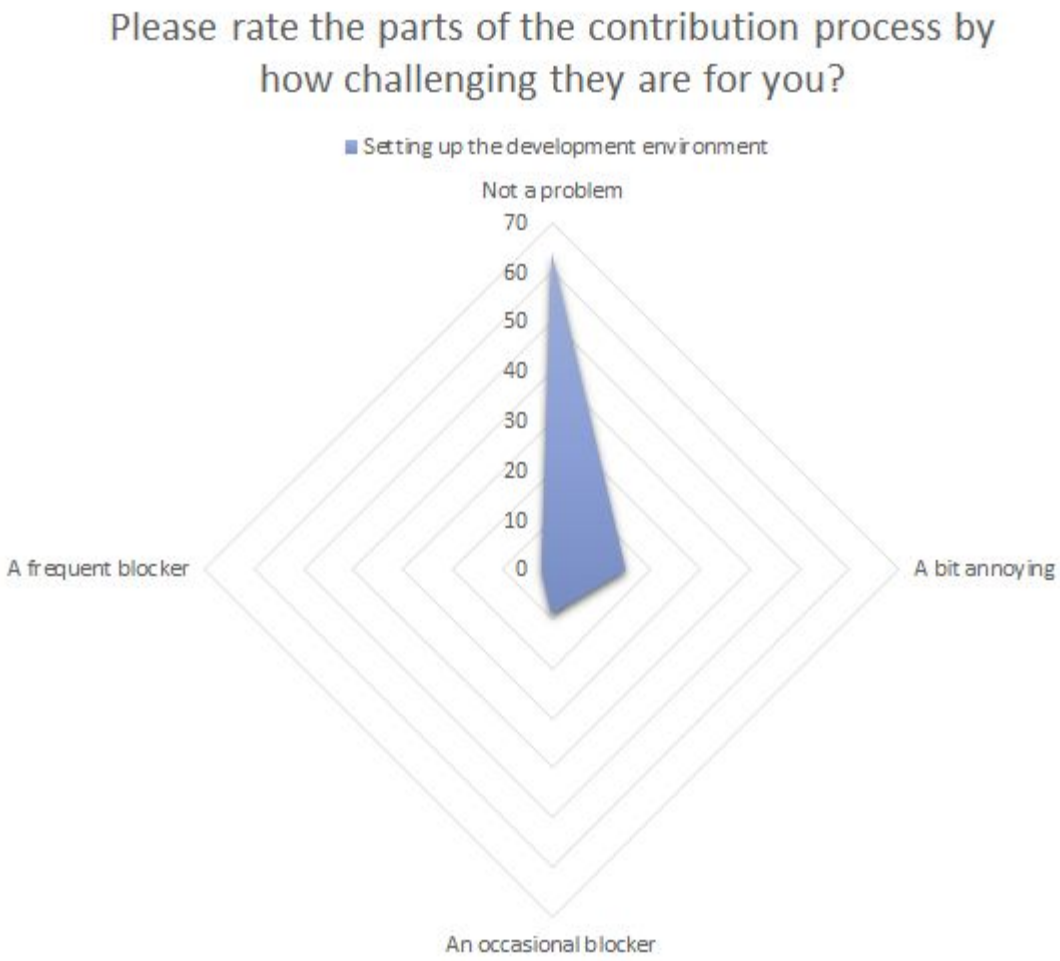

Please rate the parts of the contribution process by how challenging they are for you?
Establishing consensus for changes
Not a problem
35
30
25
20
15
10
5
0
A bit annoying
An occasional blocker
A frequent blocker

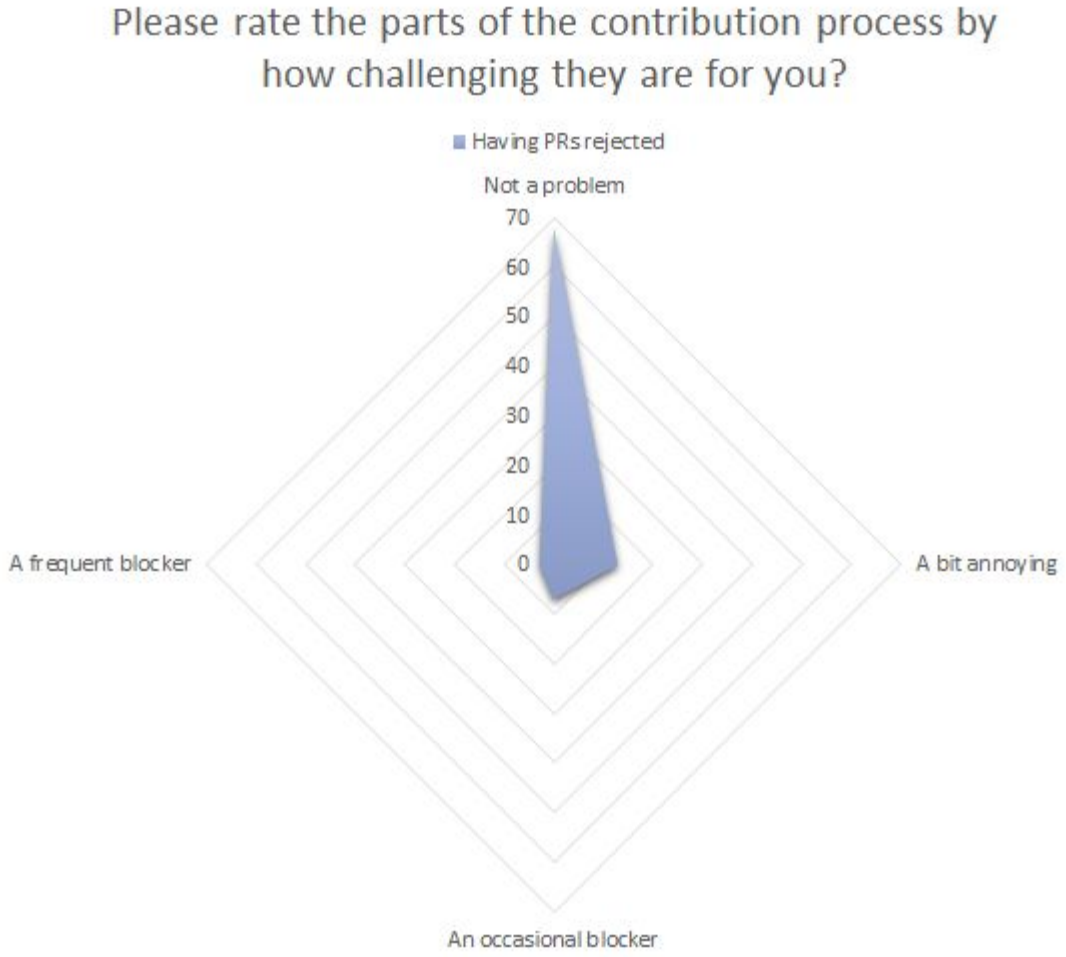

Establishing consensus for changes
3 to 5 year
2 to 3 year
1 to 2 year
< 1 year
0%
20%
40%
60%
80%
100%
A bit annoying
An occasional blocker
A frequent blocker
Not a problem
No Answer

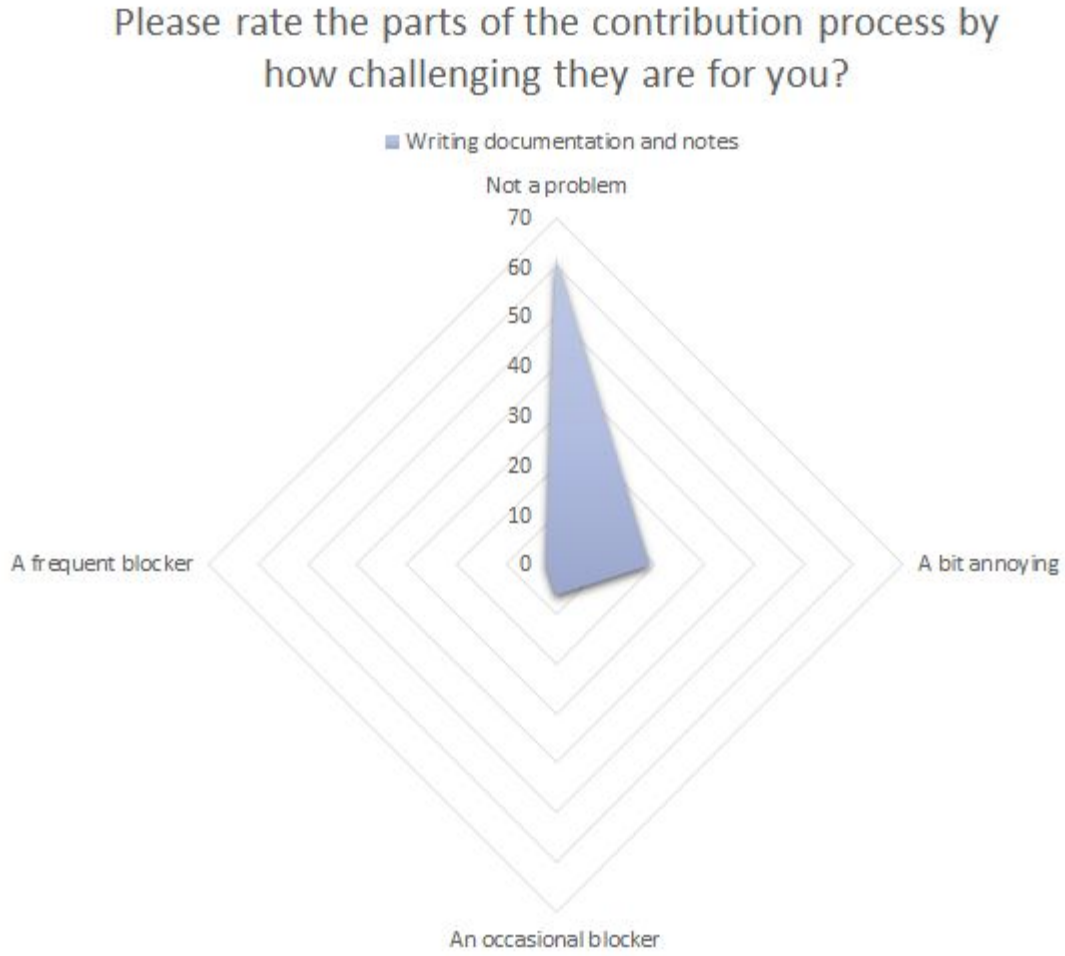

Please rate the parts of the contribution process by how challenging they are for you?
Our CI, labels, and crafted customized automation
Not a problem
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker

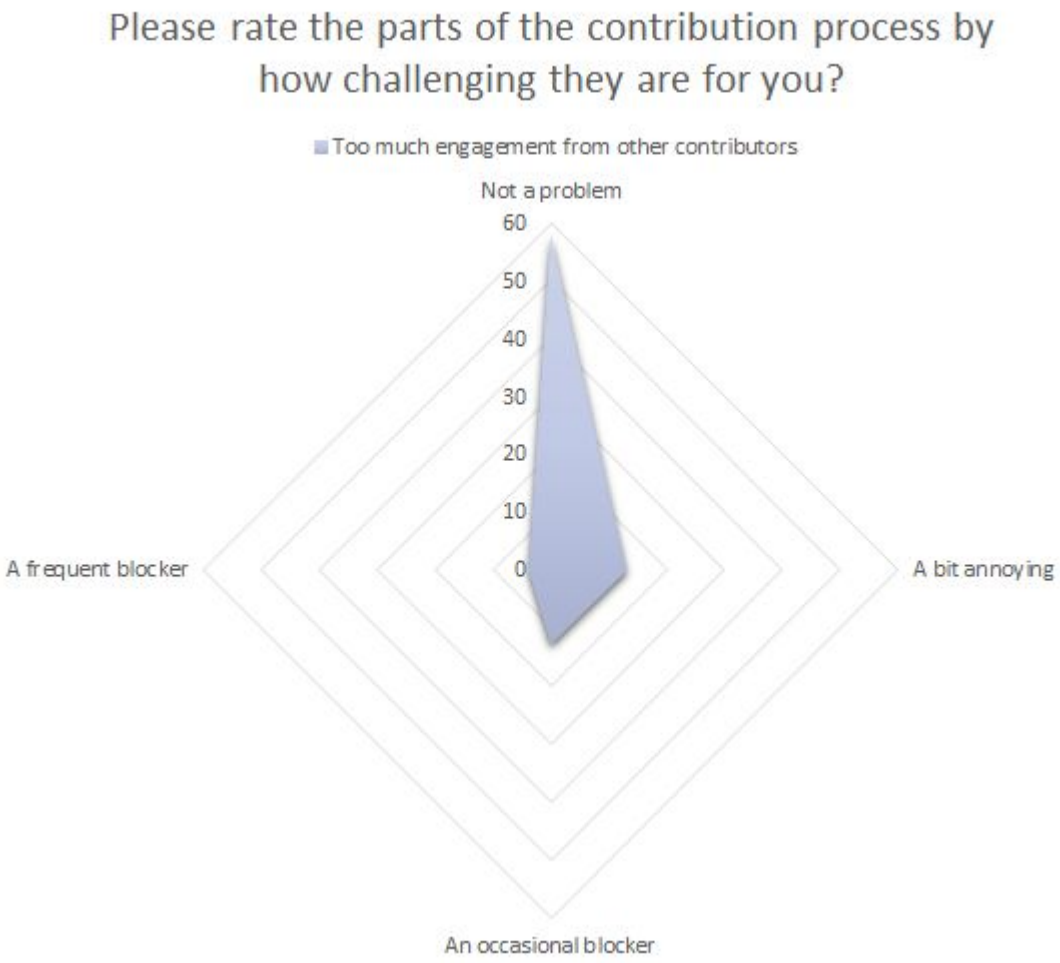

Please rate the parts of the contribution process by how challenging they are for you?
Debugging test failures
Not a problem
35
30
25
20
15
10
5
0
A bit annoying
An occasional blocker
A frequent blocker

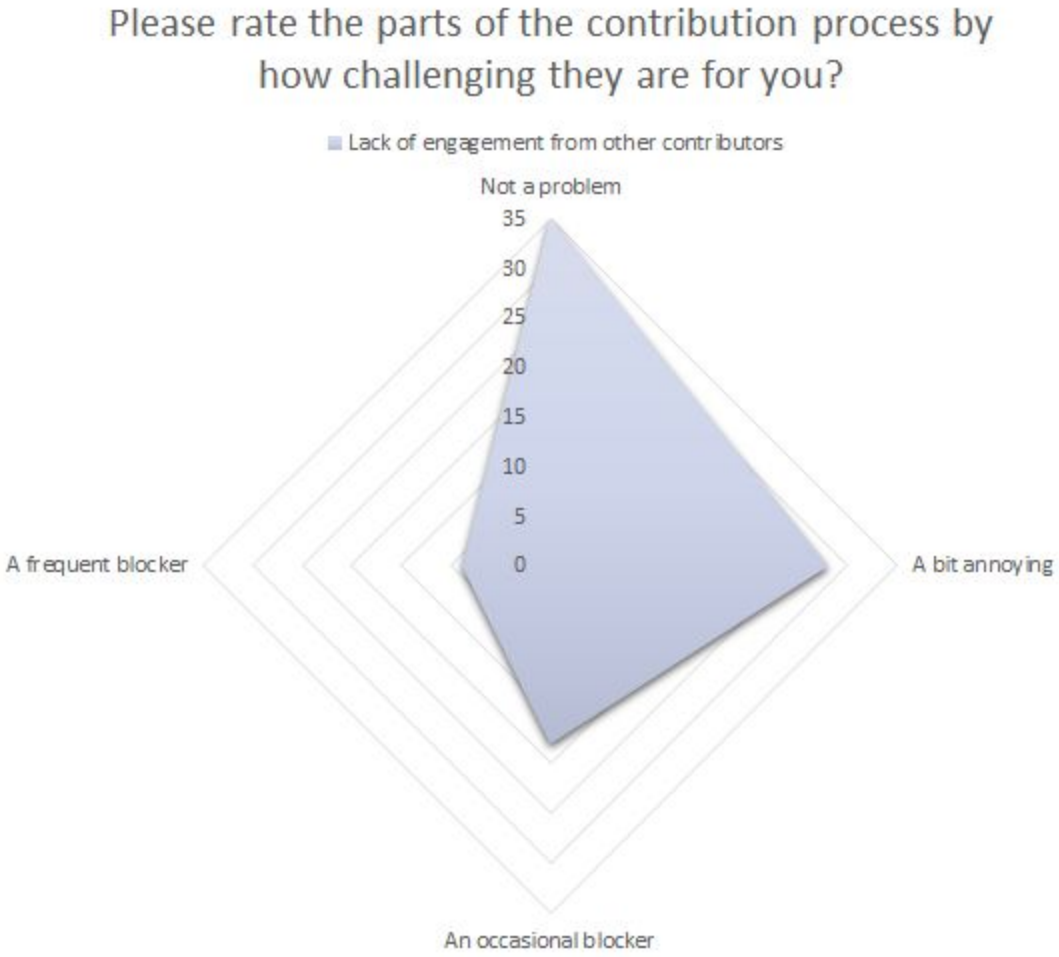

Please rate the parts of the contribution process by how challenging they are for you?
Finding appropriate issues to work on
Not a problem
50
45
40
35
30
25
20
15
10
5
0
A bit annoying
An occasional blocker
A frequent blocker

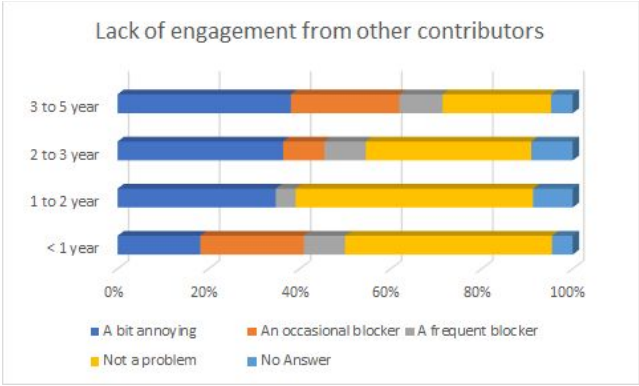

Please rate the parts of the contribution process by how challenging they are for you?
Setting up the development environment
Not a problem
70
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker


Please rate the parts of the contribution process by how challenging they are for you?
Having PRs rejected
Not a problem
70
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker


Please rate the parts of the contribution process by how challenging they are for you?
Writing documentation and notes
Not a problem
70
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker


Please rate the parts of the contribution process by how challenging they are for you?
Too much engagement from other contributors
Not a problem
60
50
40
30
20
10
0
A bit annoying
An occasional blocker
A frequent blocker


Please rate the parts of the contribution process by how challenging they are for you?
Lack of engagement from other contributors
Not a problem
35
30
25
20
15
10
5
0
A bit annoying
An occasional blocker
A frequent blocker

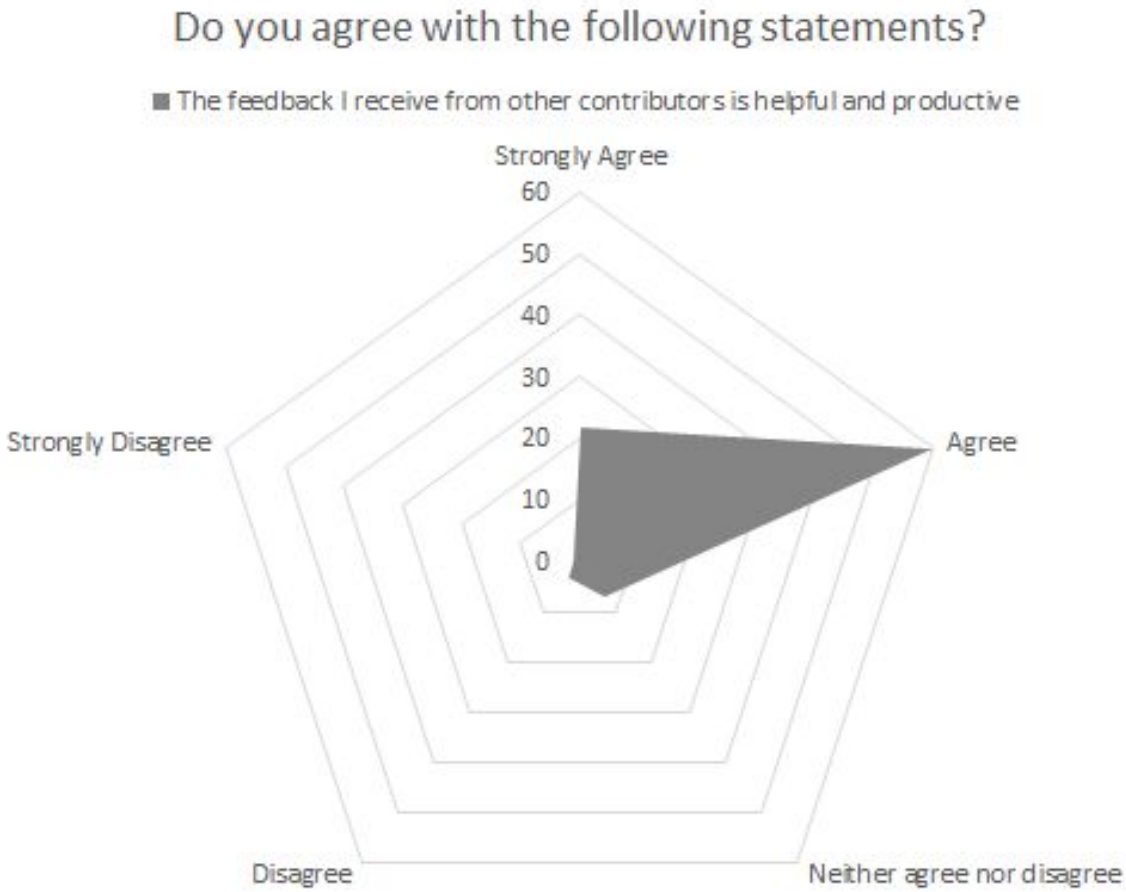

Lack of engagement from other contributors
3 to 5 year
2 to 3 year
1 to 2 year
< 1 year
0%
20%
40%
60%
80%
100%
A bit annoying
An occasional blocker
A frequent blocker
Not a problem
No Answer

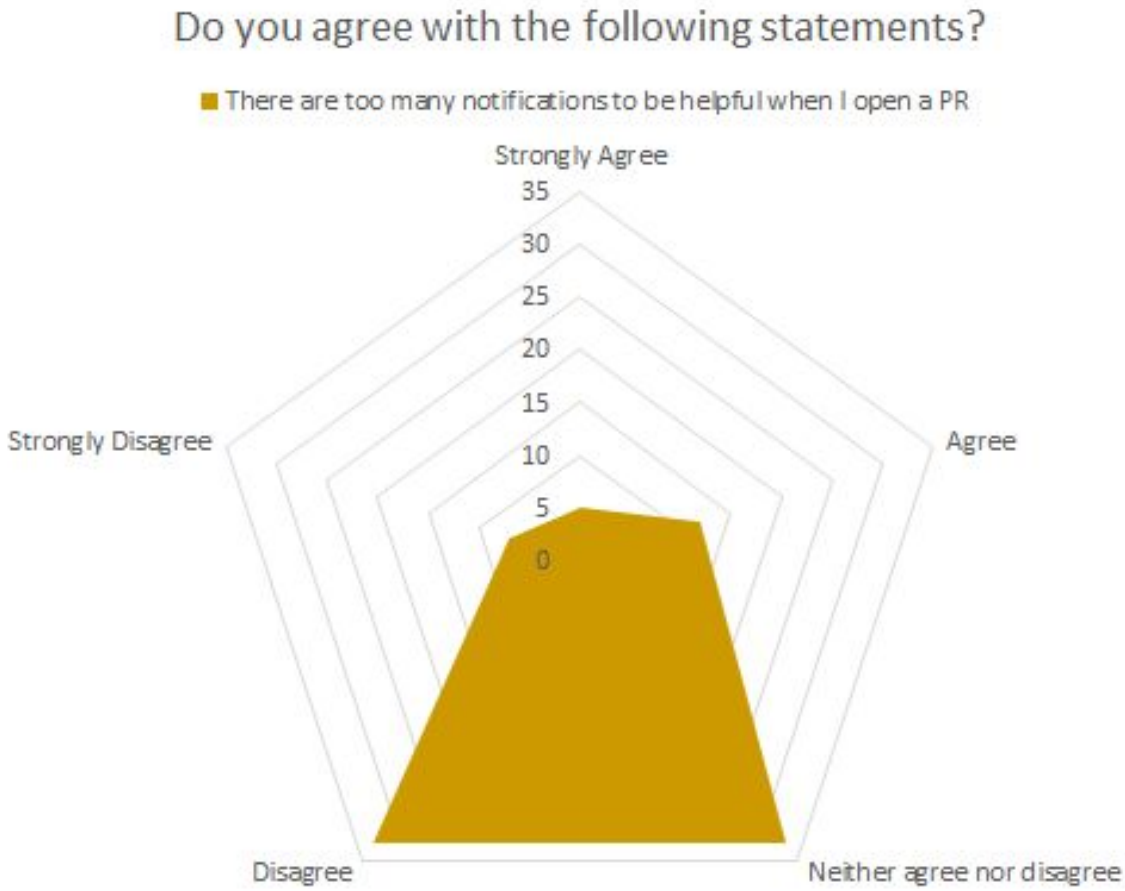
Other blockers

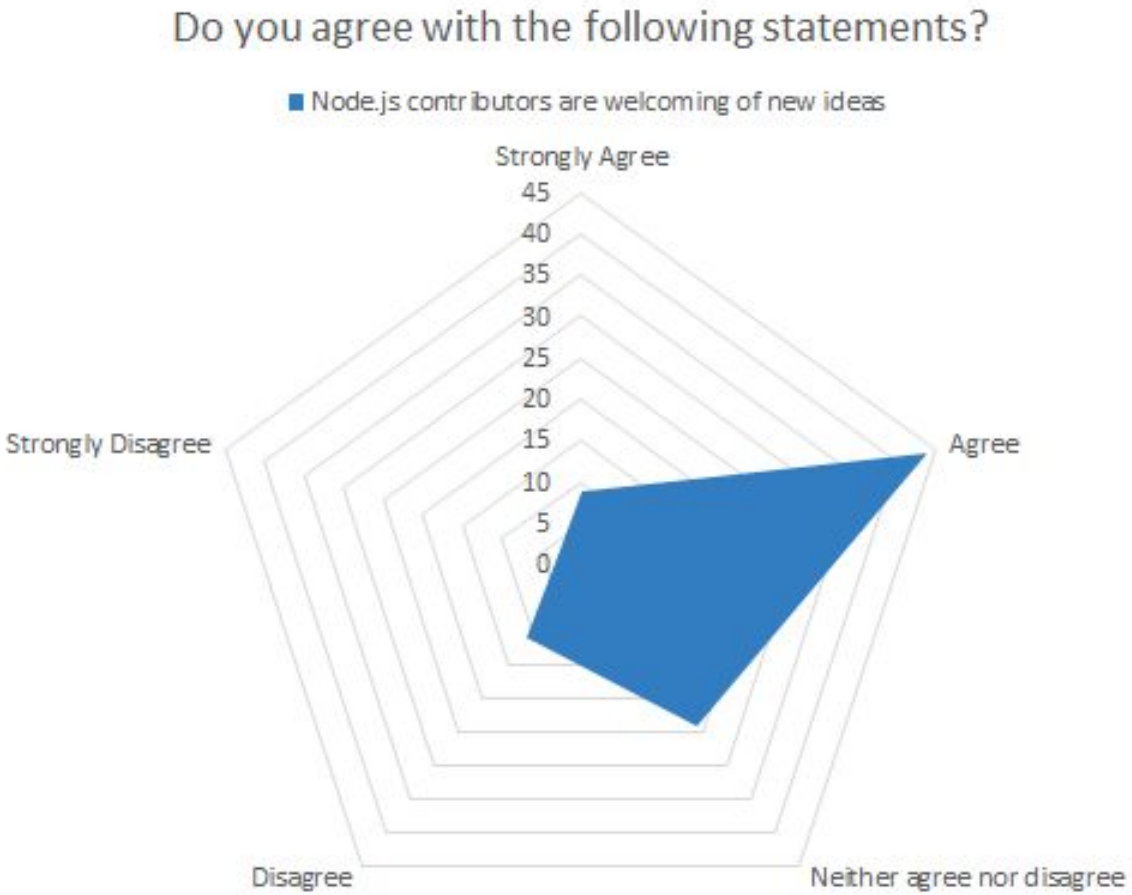
- We use IRC for communications which really sucks! No really!
- Some members game the consensus process. Instead of seeking consensus in good faith, it becomes a standoff to see who will back down first.
- Community itself
- Just a comment on the question above: while the CI is 4 for me (due to flakiness and infra issues), I'm mostly happy with the state of labels and automation in general (it could be easier to contribute to github-bot, that would improve automation). Since I believe the CI problems out weight the state of labels and automation I chose "frequent blocker" for this one.
- Project tooling overhead outweighs benefit of making small fixes in Core. Friction often not worth the hassle and makes me pursue alternative solutions/workarounds. Could be by design though, and helpful or not.
- Beyond "good first issue" stuff, there's a distinct lack of any cohesive roadmap or wishlist to guide future work at a deeper or more complex level.
- Code sometimes is quite cryptic
- Things move too fast at times; progress is prioritized over caution.
- Having PRs ignored
- The big one is the review process.
- Bikeshedding. We are too obsessed by making things perfect in the first run
- Node.js' lack of using native features in GitHub, like merging in nodejs/node (as opposed to closing). Additionally, our approach to automation as an image: https://imgur.com/a/56j37ol
- I think a pure GitHub merge process without special tooling would make it a lot more approachable.

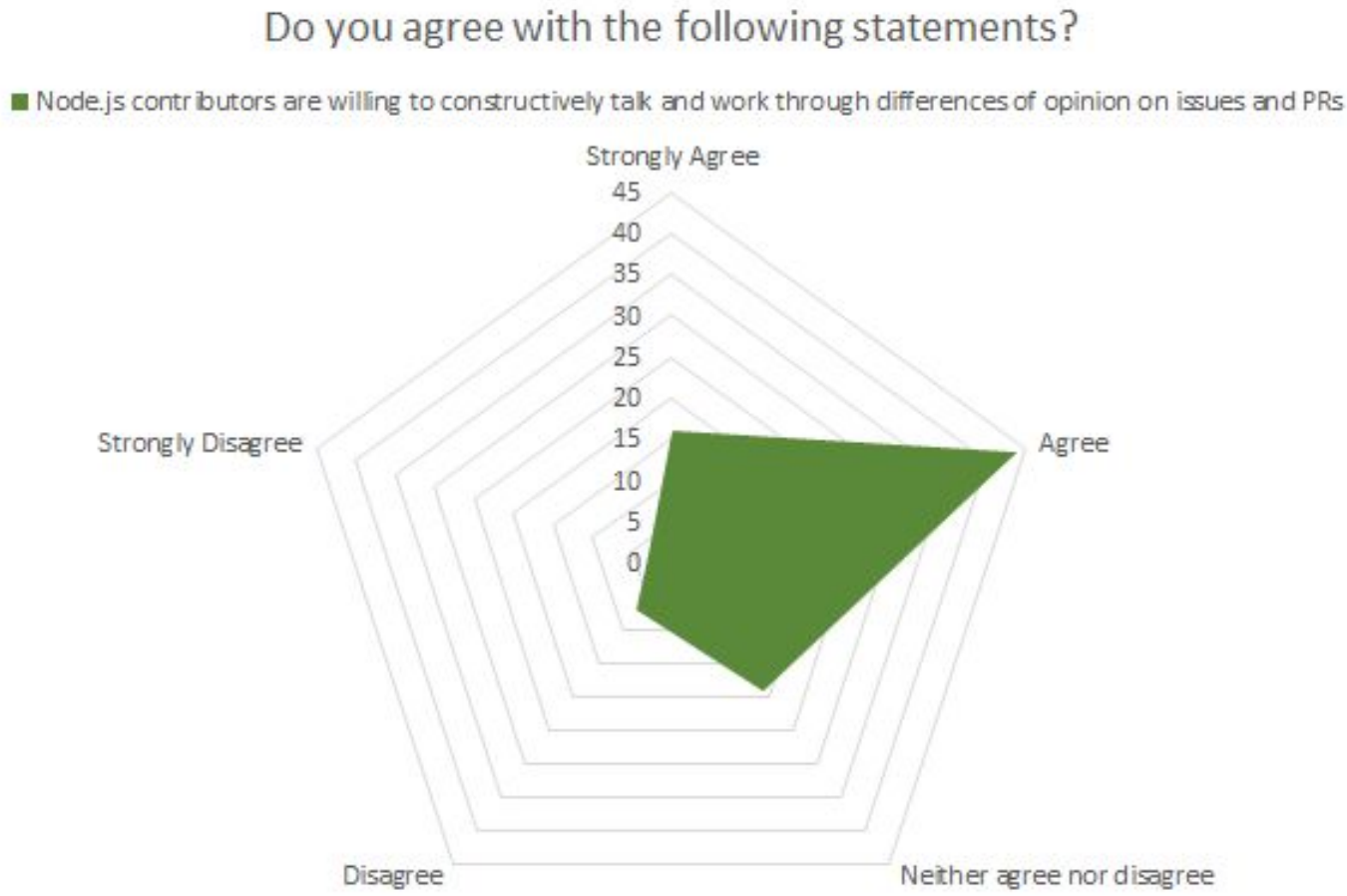
Other blockers (continued)

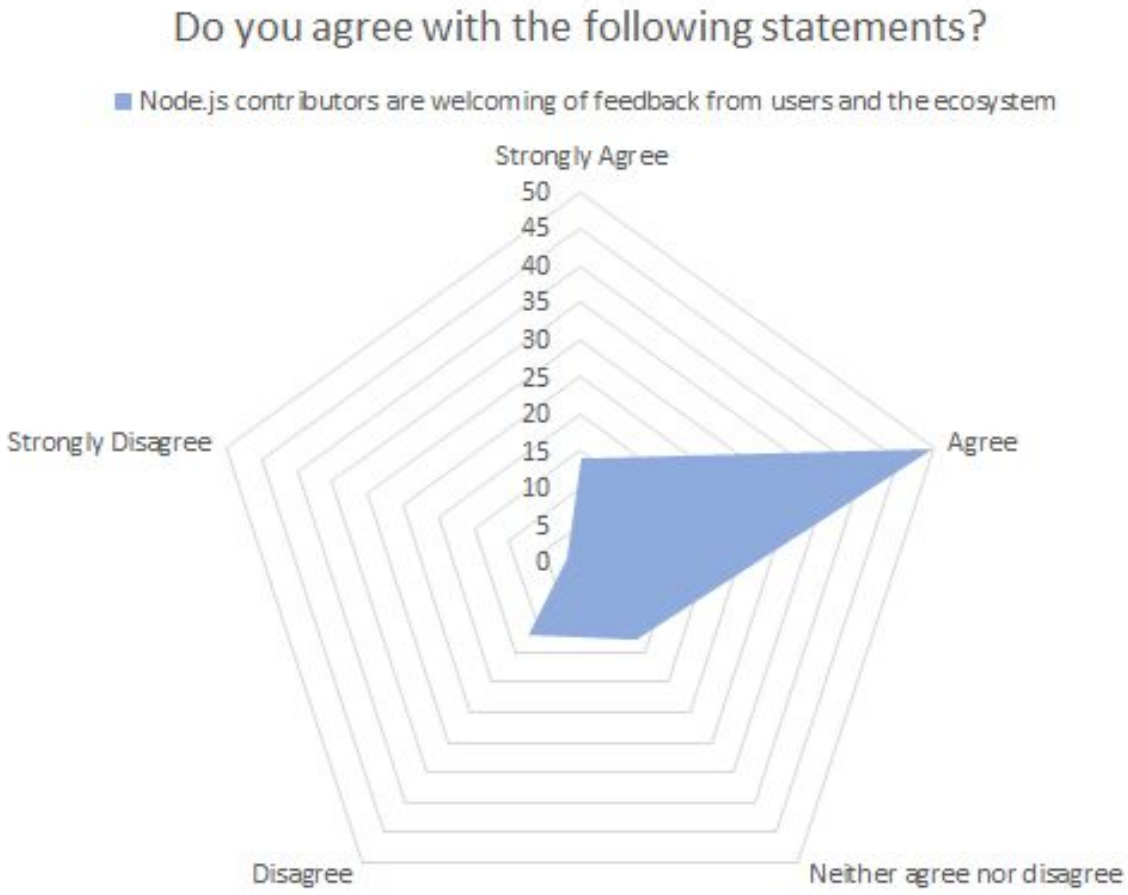
- "There are a few individuals that are extremely noisy and write complex comments to grasp. Their language is sophisticated and they are hard to follow for non-native english speakers.
- They are making Node.js seem an exclusive club of people who have a lot of time to contribute."
- Having a proper channel for communication like Slack, though we have IRC but it is not that much easy to find and colloborate.
- Having a mentor would help me contribute more as I constantly feel stupid asking things.
- Having to ask someone to re-run CI is annoying and feels like an unnecessary burden for both sides.
- Source code/architecture/design documentation
- really hard to understand how to contribute.
- A lack of people who have enough experience with a problem to be available to deeply discuss the details with in considering new work.
- Being dragged into unnecessary discussions that don't necessarily cover the primary concerns at hand.
- Limited appreciation for good contributions to the project breaks the sense of feeling like an appreciated part of a community.
- Financial restrictions when it comes to donating time to fix issues that may be related to work but not necessarily on ones own personal critical path, especially when the issues being fixed primarily benefit large companies.

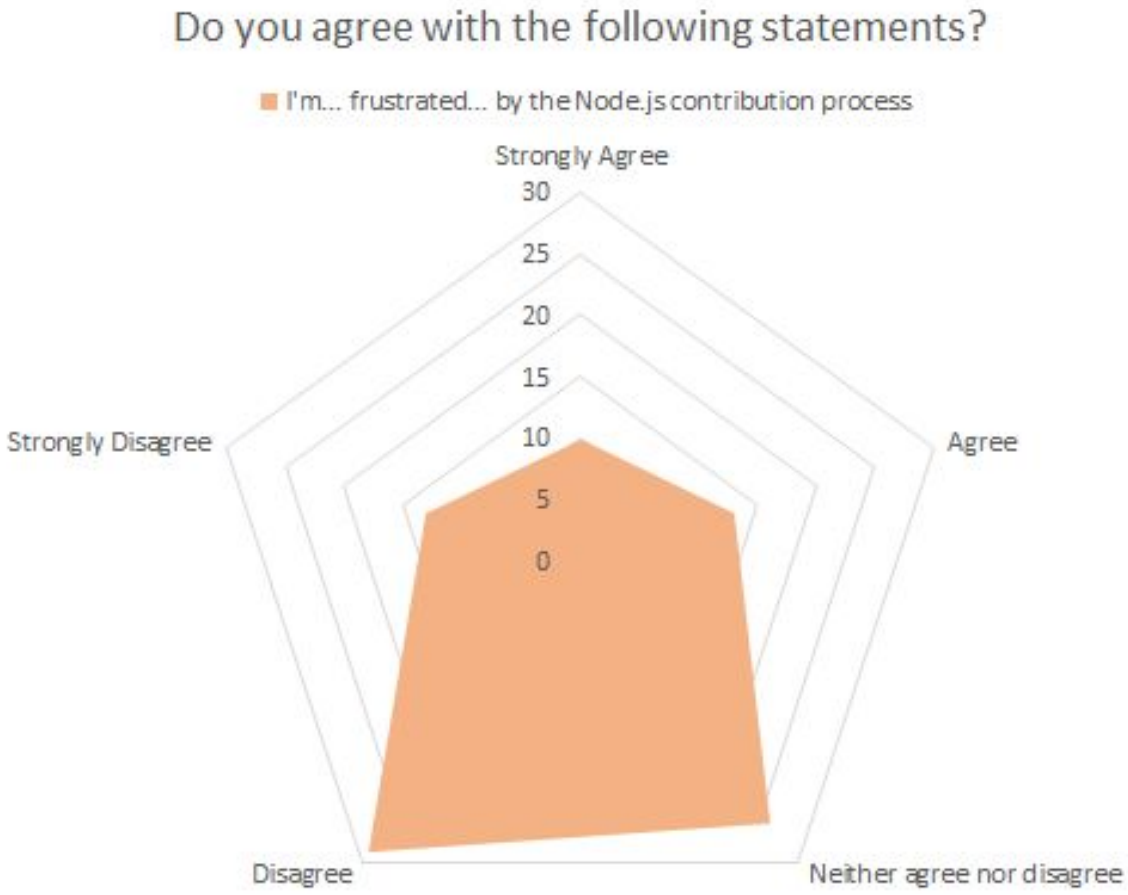

Do you agree with the following statements?
The feedback I receive from other contributors is helpful and productive
Strongly Agree
60
50
40
30
20
10
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

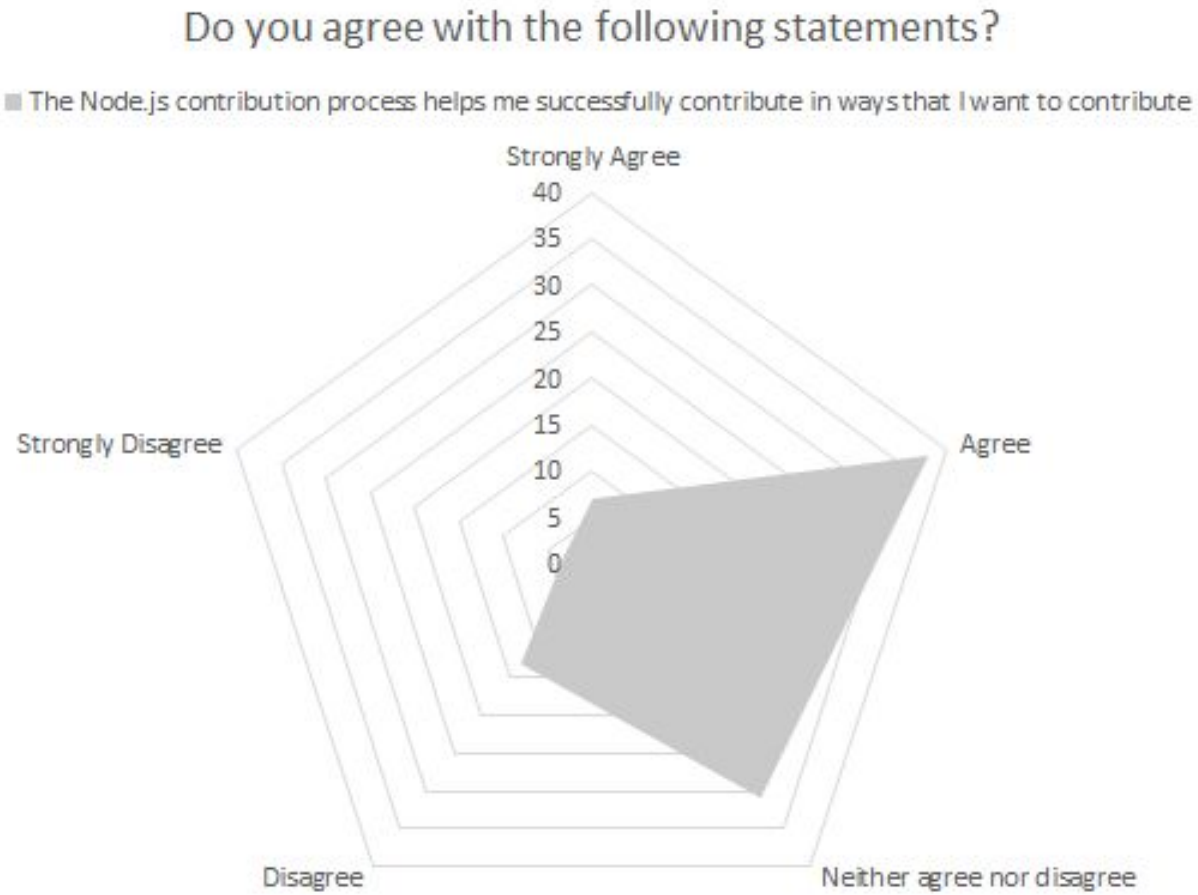

Do you agree with the following statements?
There are too many notifications to be helpful when I open a PR
Strongly Agree
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

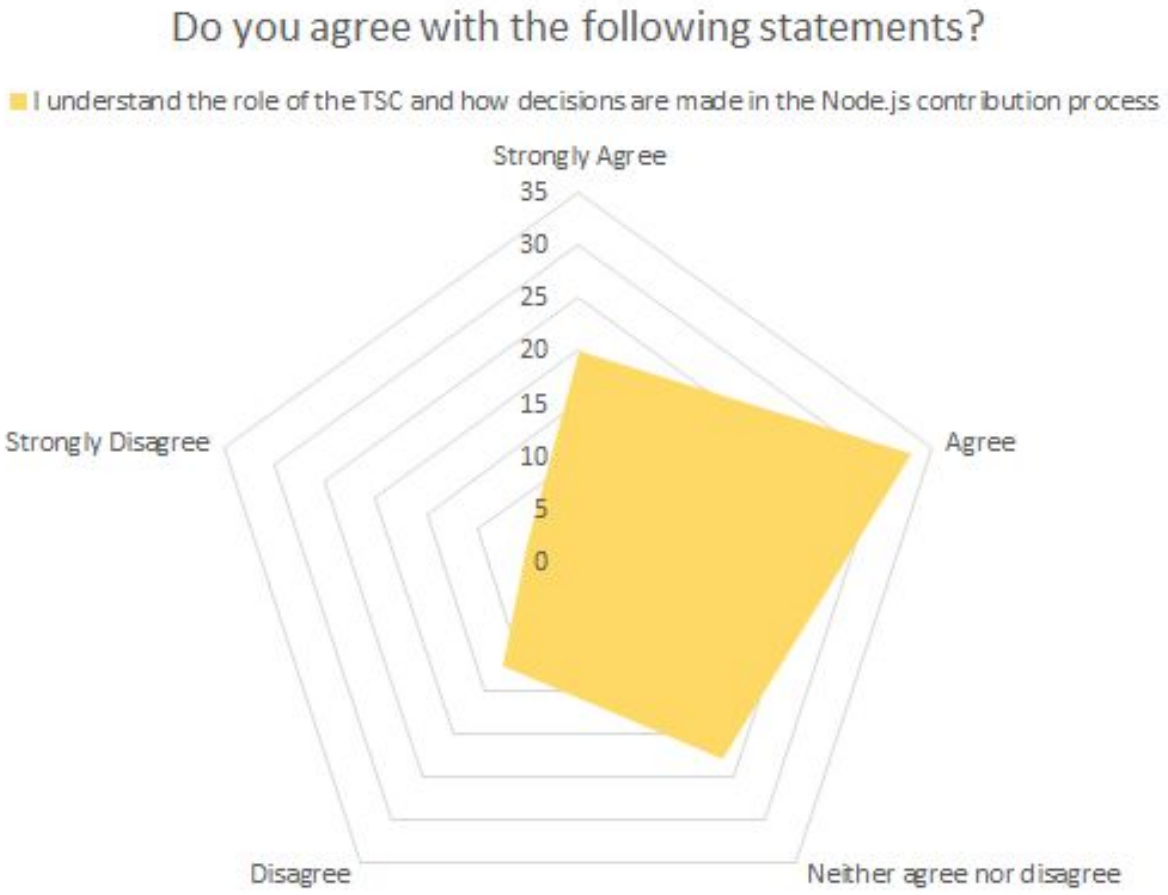

Do you agree with the following statements?
Node.js contributors are welcoming of new ideas
Strongly Agree
45
40
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree


Do you agree with the following statements?
Node.js contributors are willing to constructively talk and work through differences of opinion on issues and PRs
Strongly Agree
45
40
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

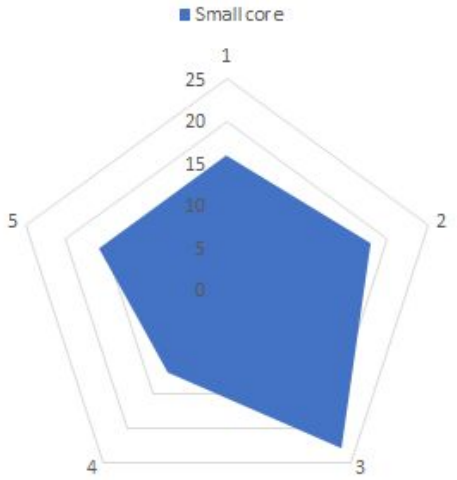

Do you agree with the following statements?
Node.js contributors are welcoming of feedback from users and the ecosystem
Strongly Agree
50
45
40
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

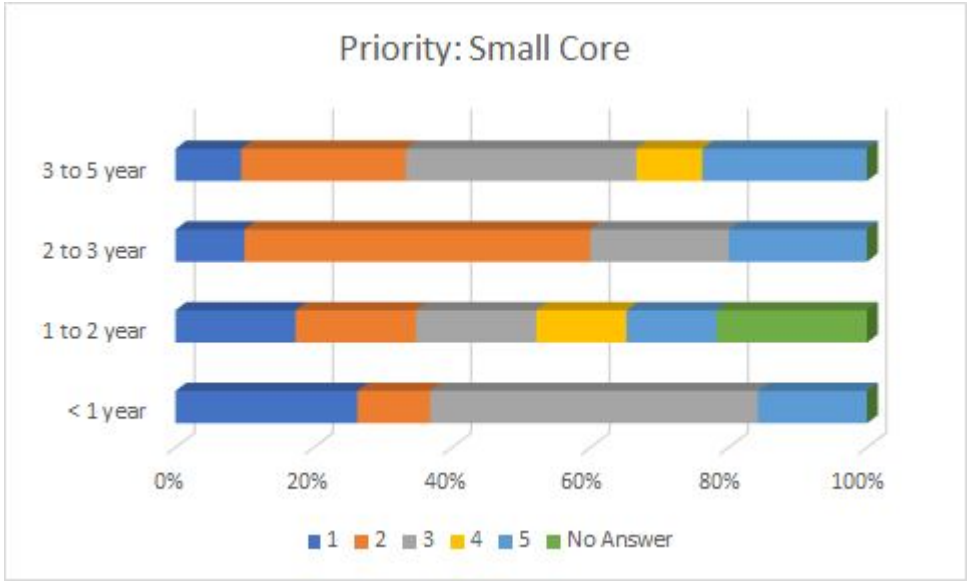

Do you agree with the following statements?
I'm... frustrated... by the Node.js contribution process
Strongly Agree
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

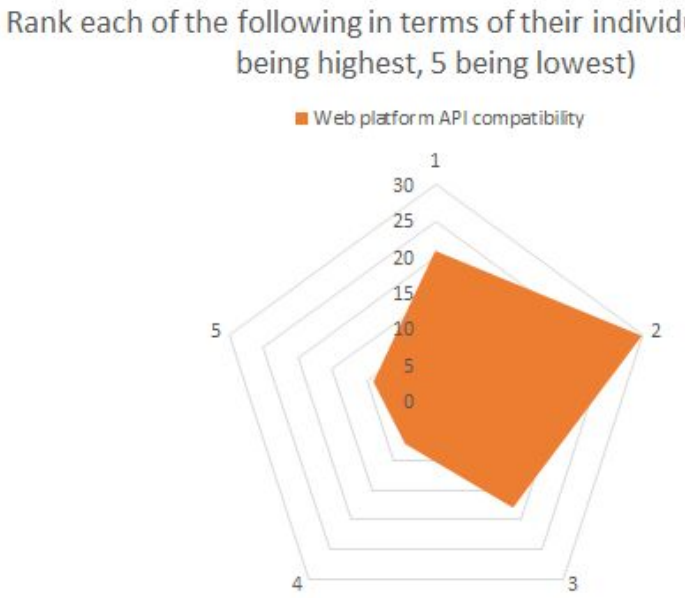

Do you agree with the following statements?
The Node.js contribution process helps me successfully contribute in ways that I want to contribute
Strongly Agree
40
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

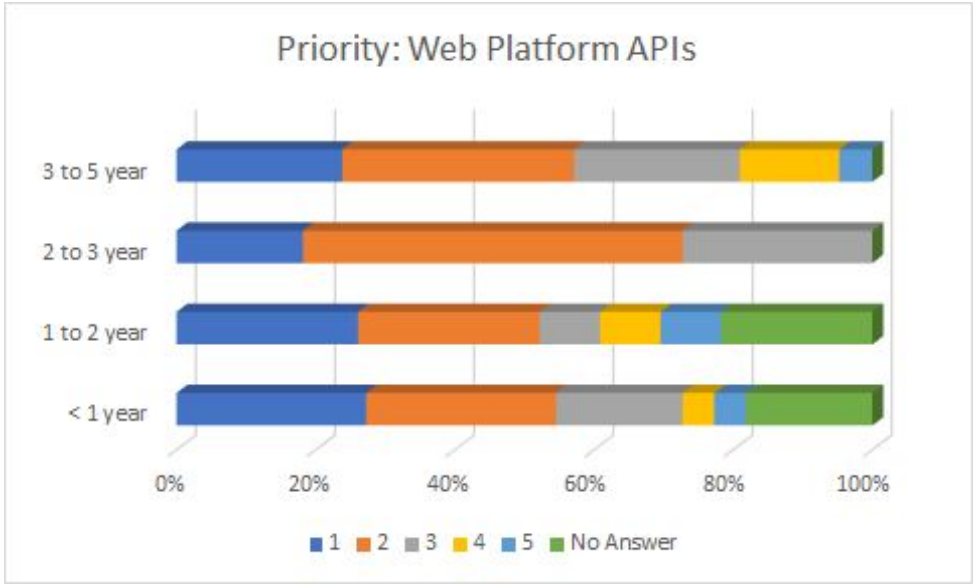

Do you agree with the following statements?
I understand the role of the TSC and how decisions are made in the Node.js contribution process
Strongly Agree
35
30
25
20
15
10
5
0
Agree
Neither agree nor disagree
Disagree
Strongly Disagree

Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)

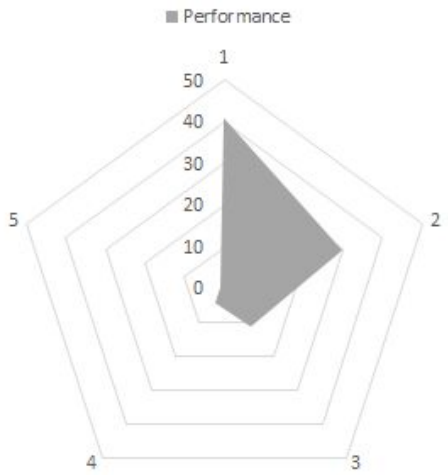


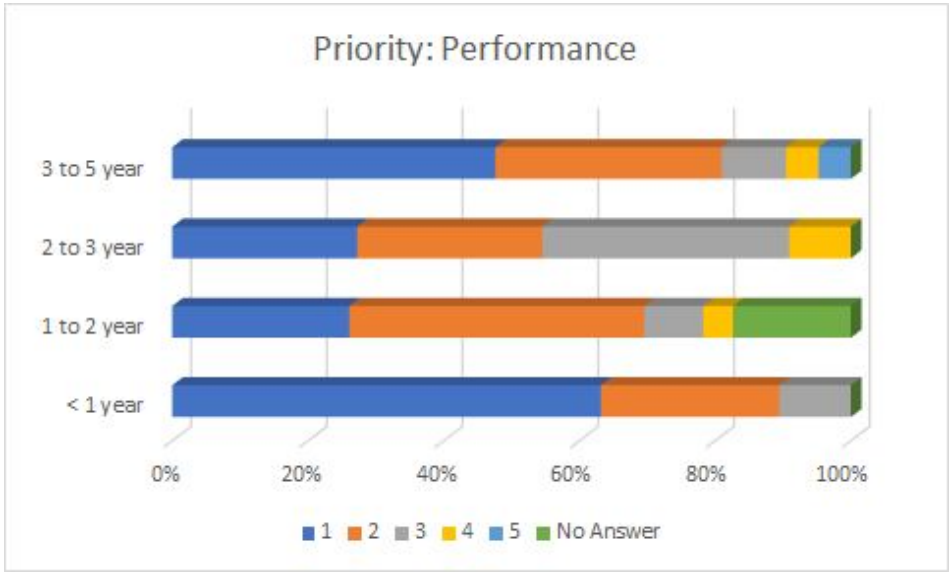



Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)
Web platform API compatibility
1
2
3
4
5
30
25
20
15
10
5
0

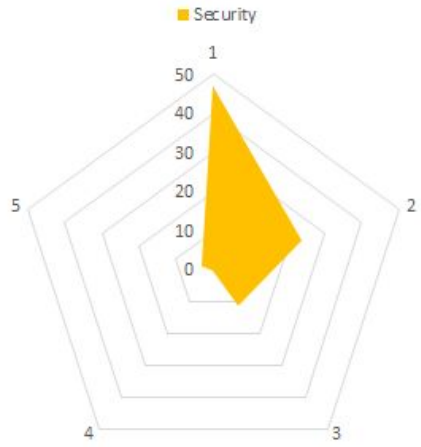

Priority: Web Platform APIs
3 to 5 year
2 to 3 year
1 to 2 year
< 1 year
0%
20%
40%
60%
80%
100%
1
2
3
4
5
No Answer

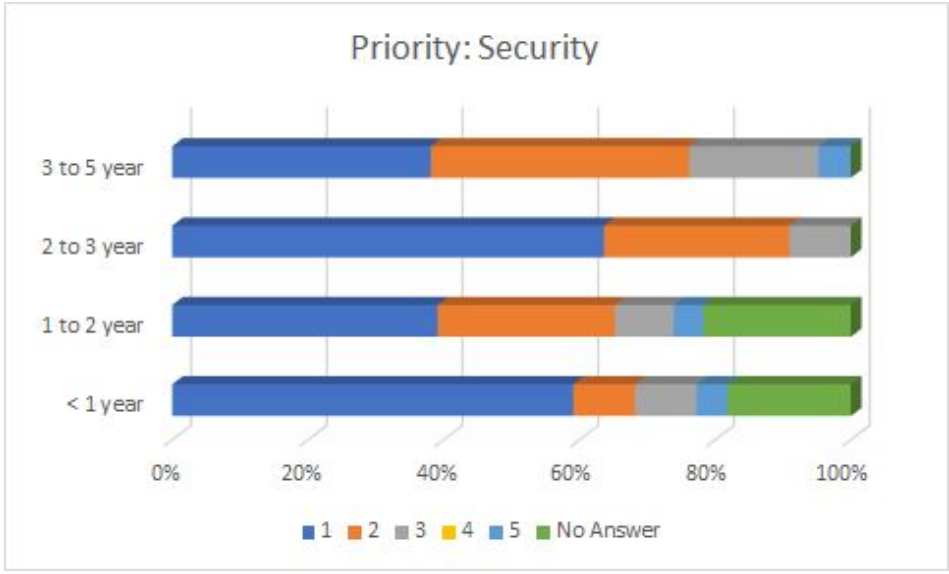
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



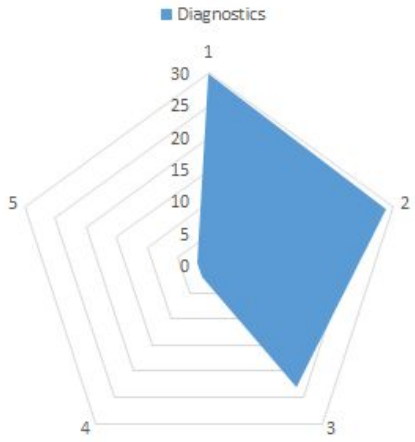


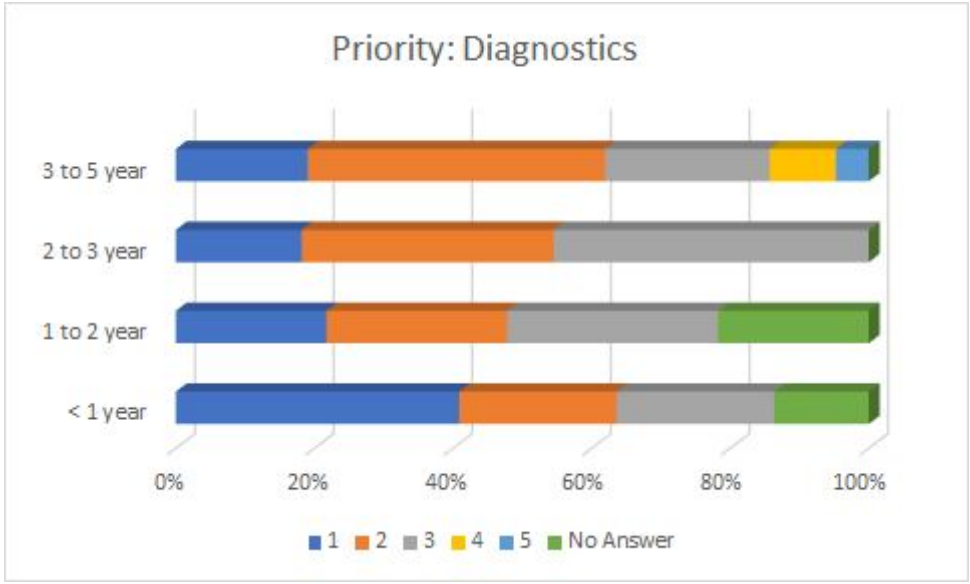
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



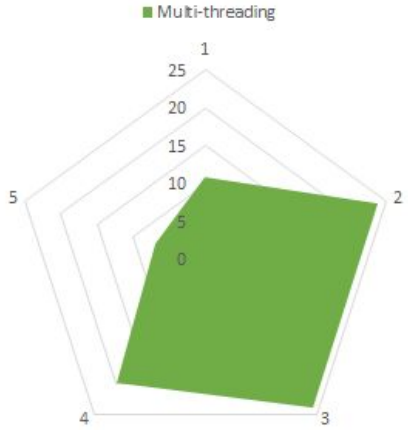


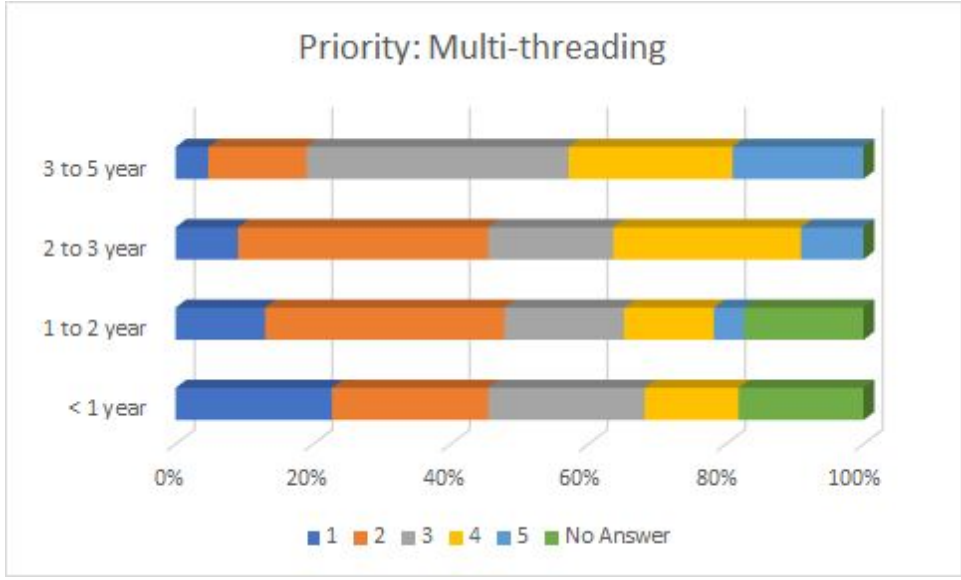
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



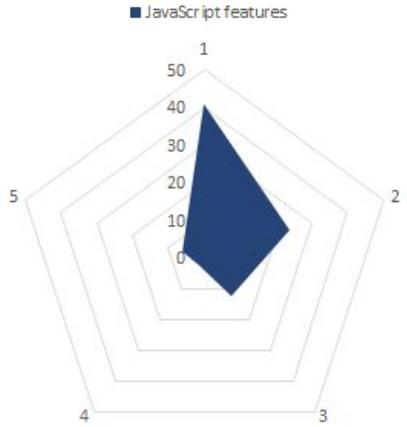


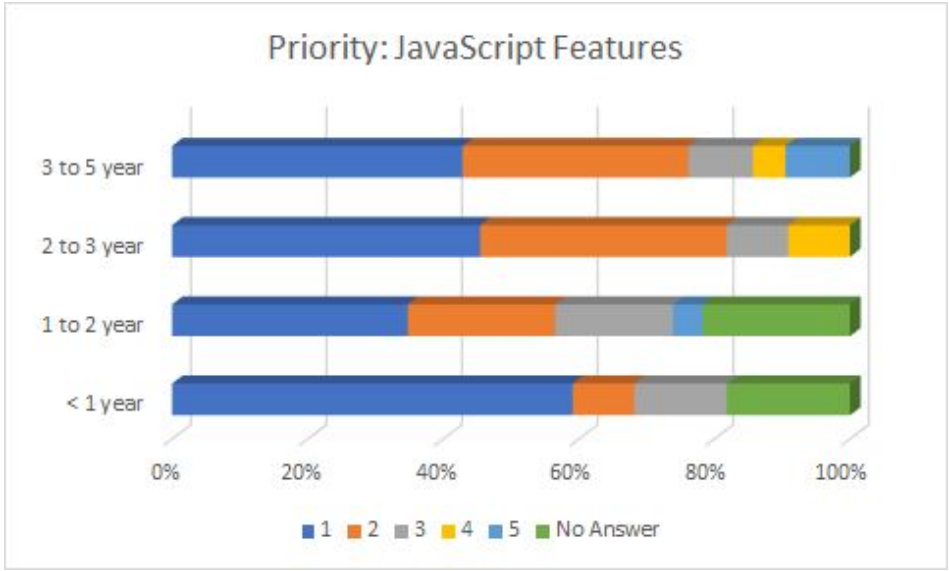
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



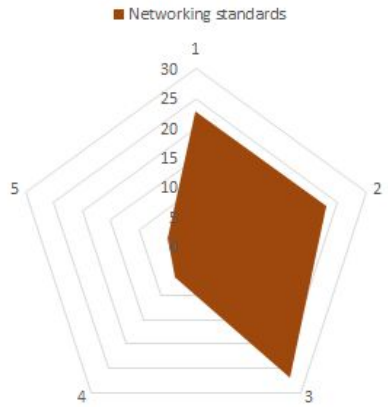


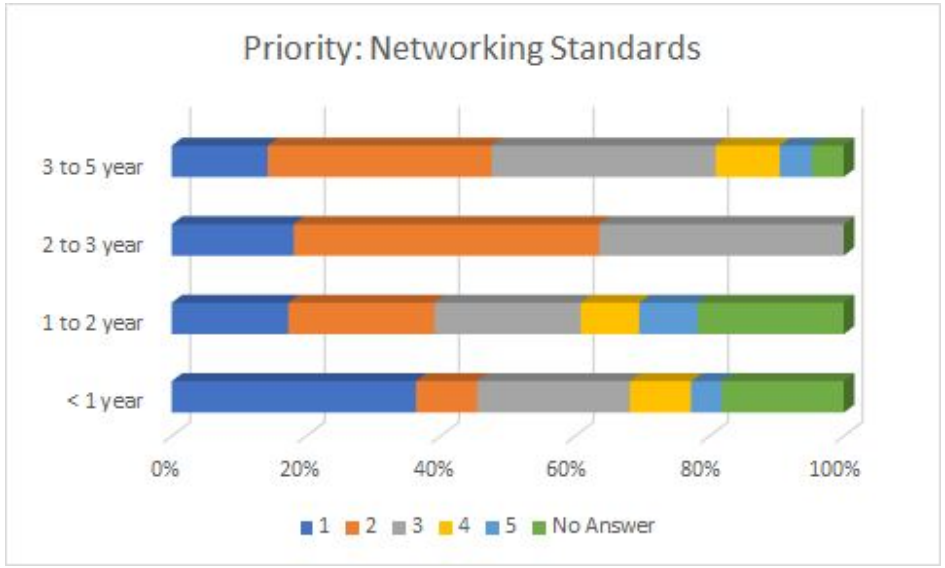
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



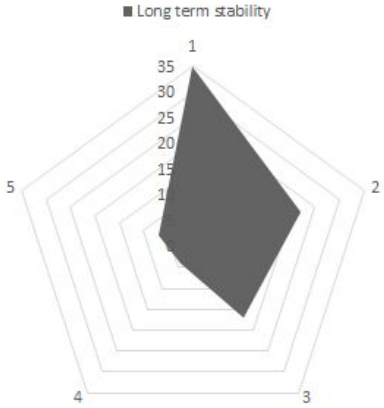


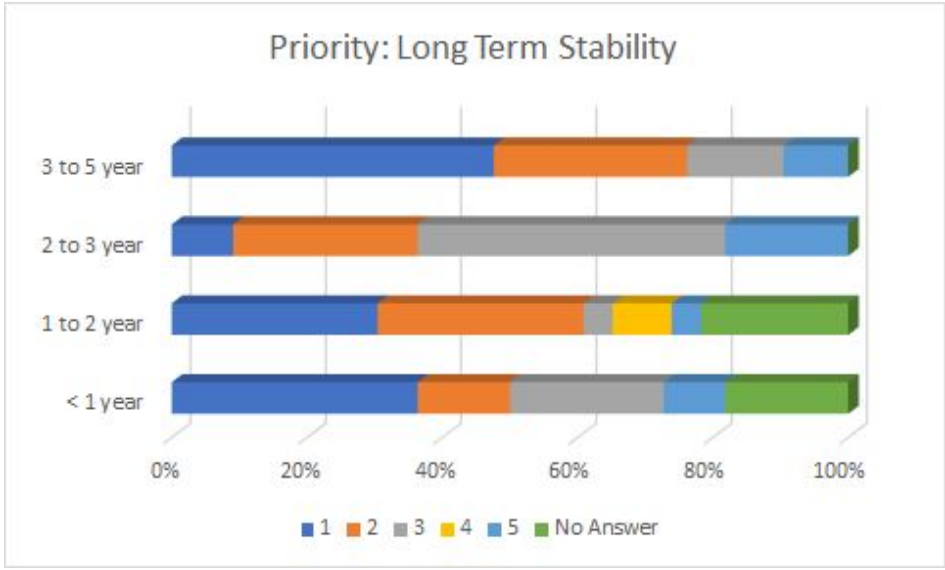
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



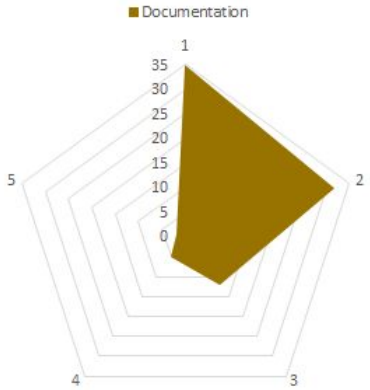


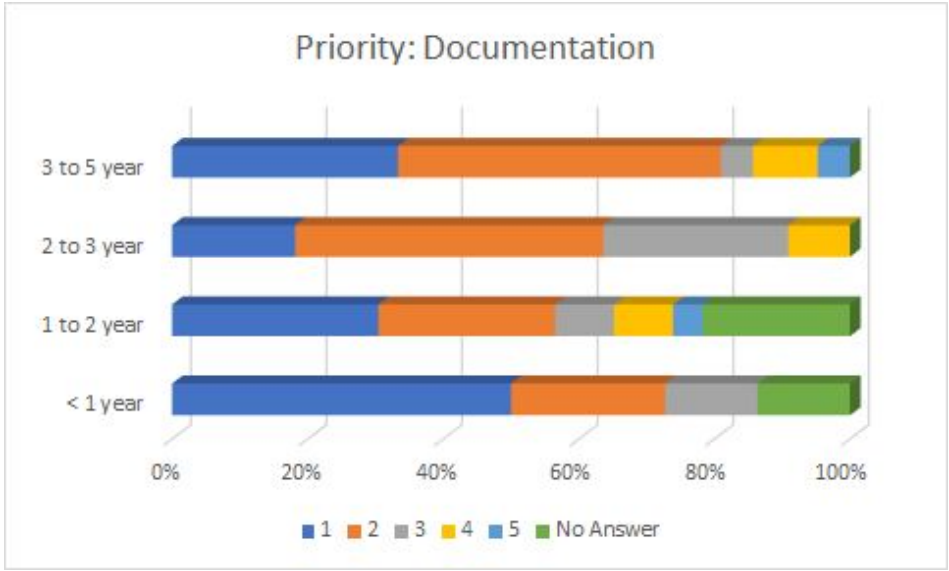
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)



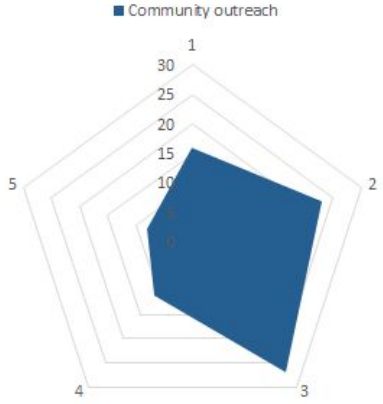


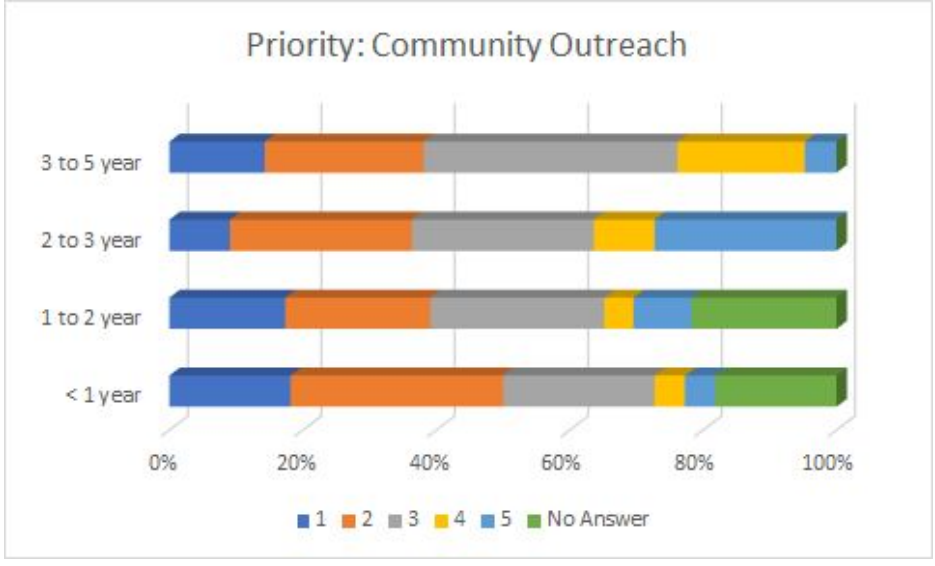
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)





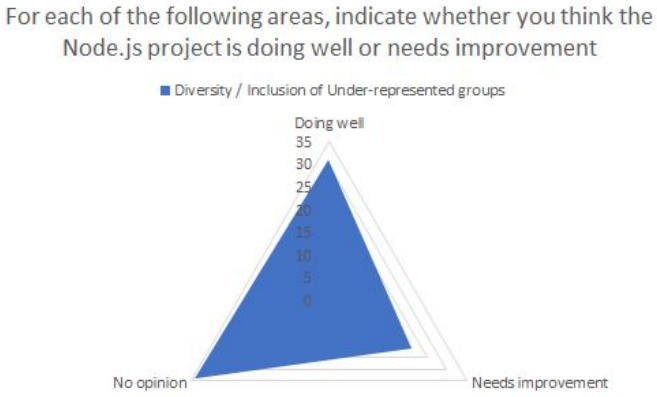
Rank each of the following in terms of their individual priority (1 being highest, 5 being lowest)

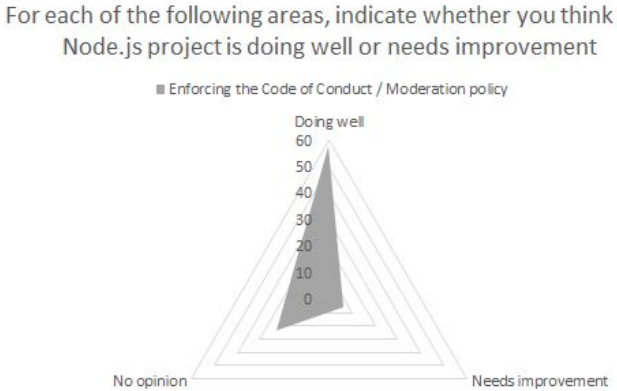




Other priority items

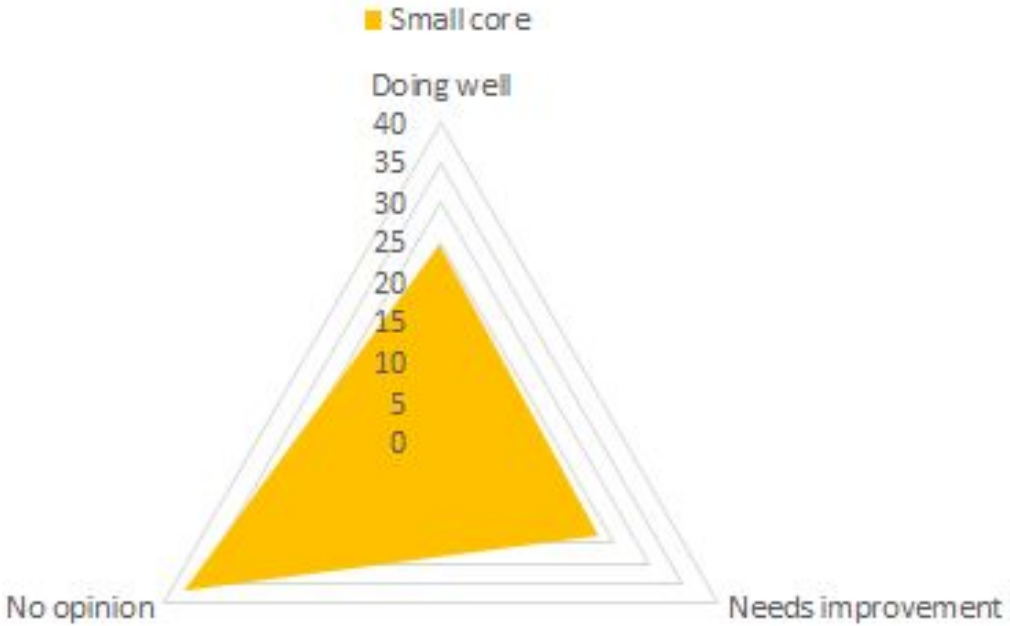
- HTTP convenience lib
- I care mostly about Node being an effective environment to write code in. So I only care about things like "small core" or "performance" in that context.
- Teachability
- modules, lowest (5)
- Removing all reliance on Python
- people writing and using tools written in Node.js, which is what most people using Node.js use Node.js for.
- Cross-platform support (CPU arch and OS) @ highest (1)
- Long term planning for the APIs. See answer to the "not listed above" question that appears after the next question.
- Making other language usage easy. users want a better typescript experience.
- A great Repl.
- Some of the design decisions in Deno are worthy of consideration. Consider built-in TypeScript support (it seems the world has spoken) and controlling the app's I/O access capabilities from the CLI.
- I/O + system permissions, transparent ESM
- Performance tooling in built
- type support, mostly typescript support.
- Better participation in standards including TC39 as well as in web standards especially as the browser scope continues to widen
- Loaders
- Having a central configuration system (including https://github.com/lorenwest/node-config or a version of it into the core) - 3 mid priority
- It would be nice to see but we have NPM for our 'wants'



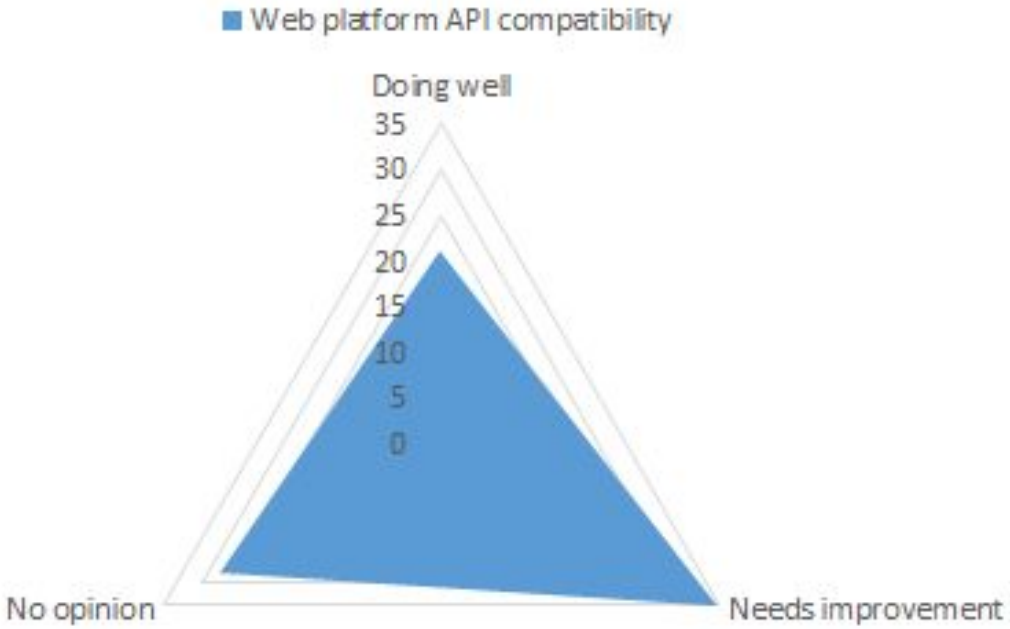


Recommendation: We need to conduct a diversity and inclusion survey across the contributor community.

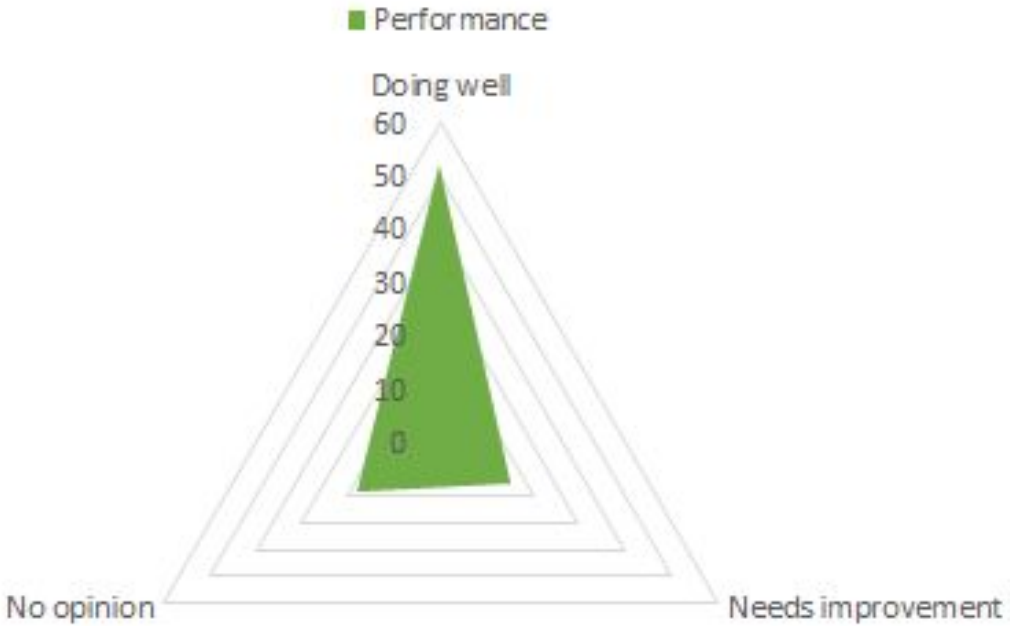
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



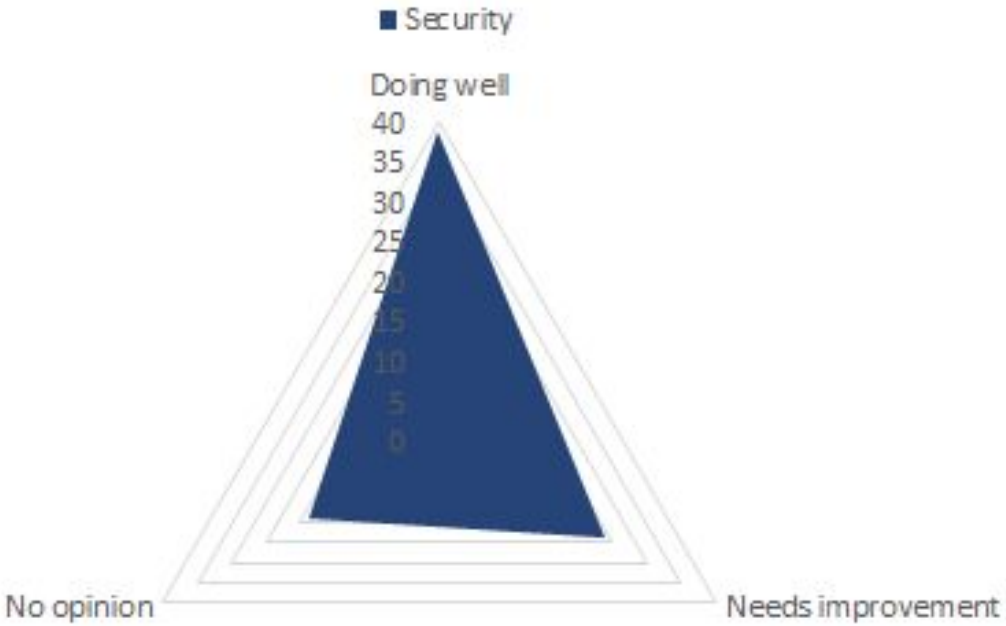
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



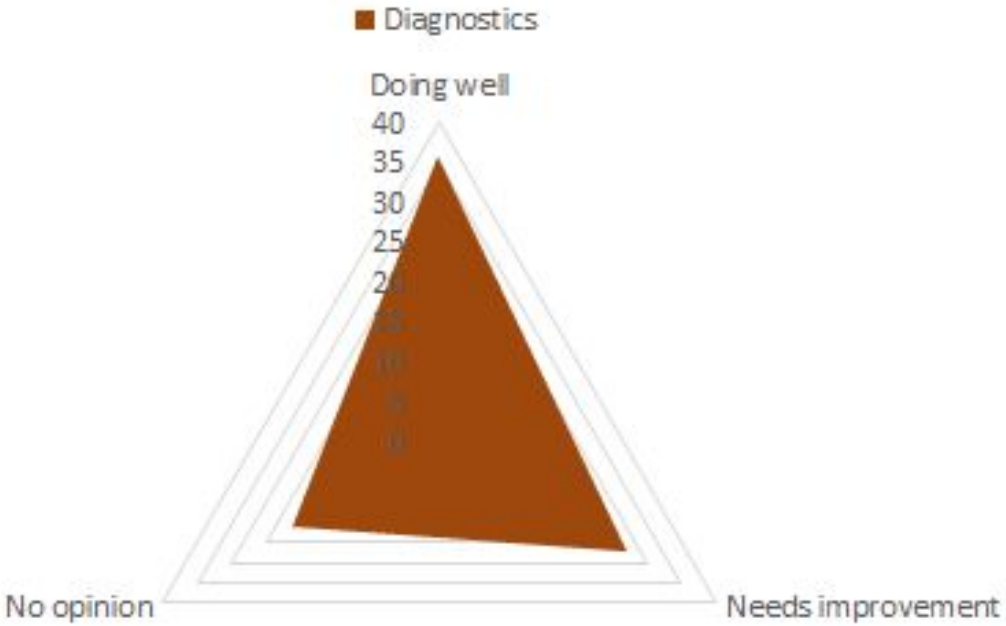
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



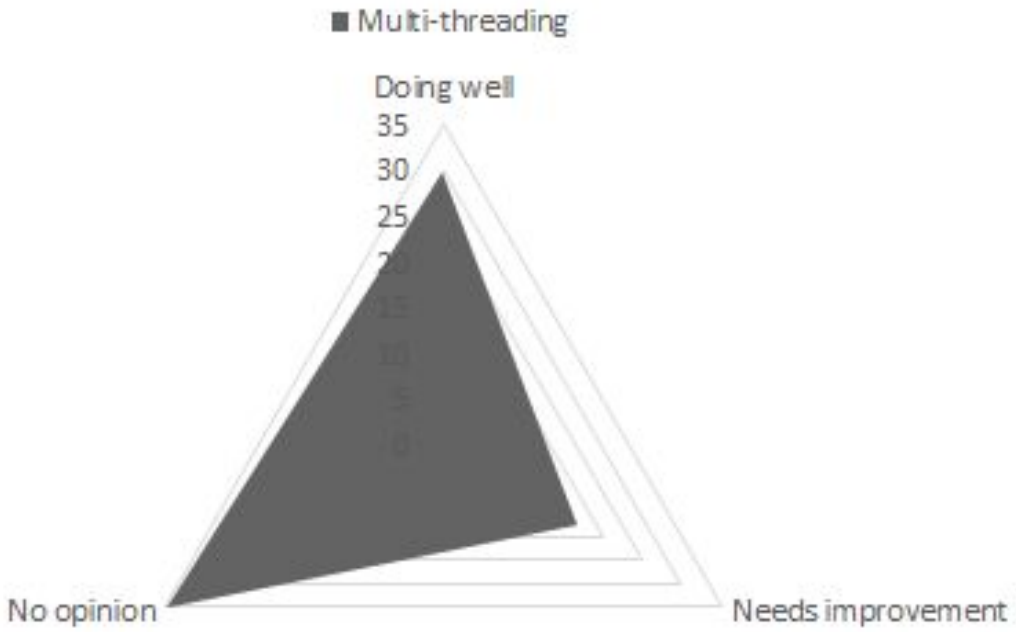
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



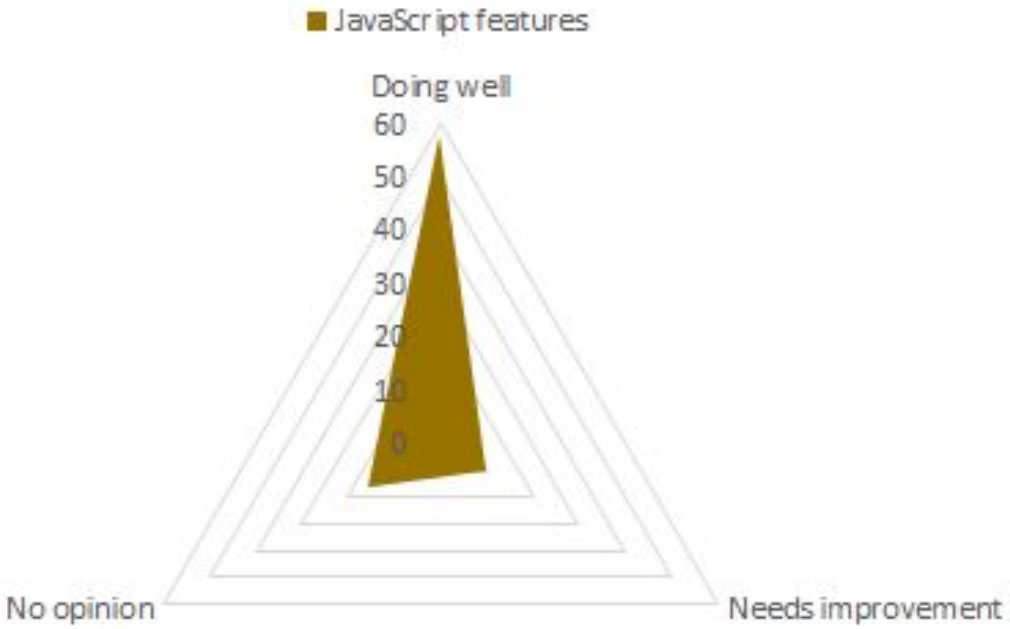
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



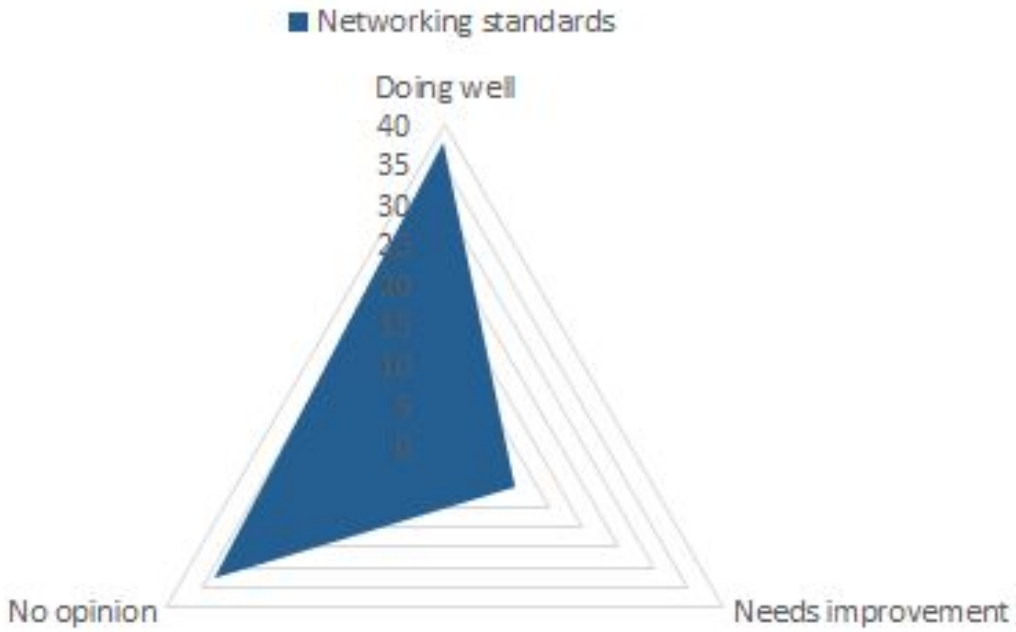
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



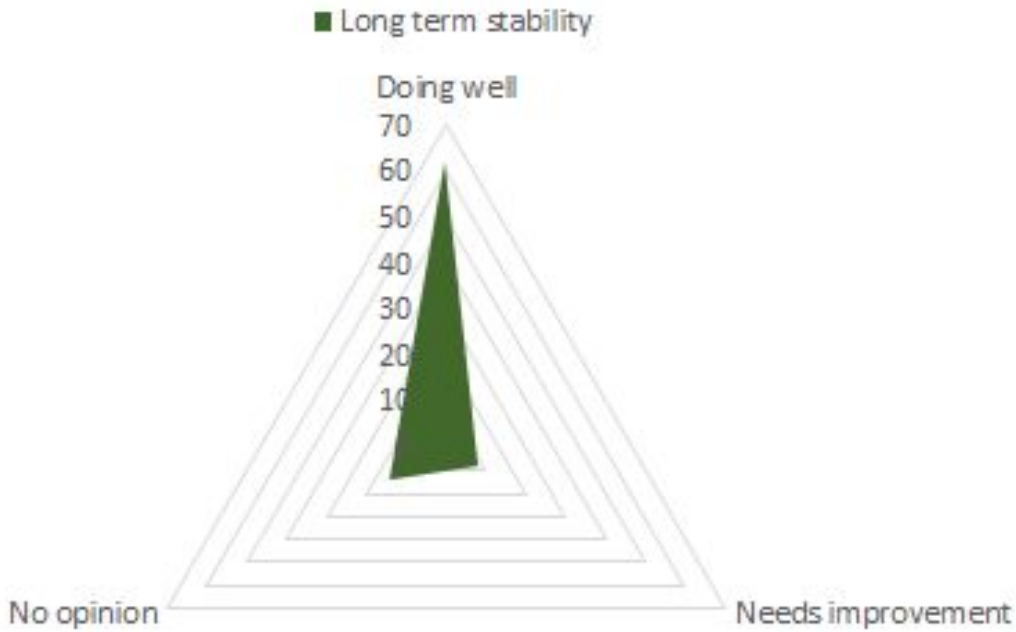
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



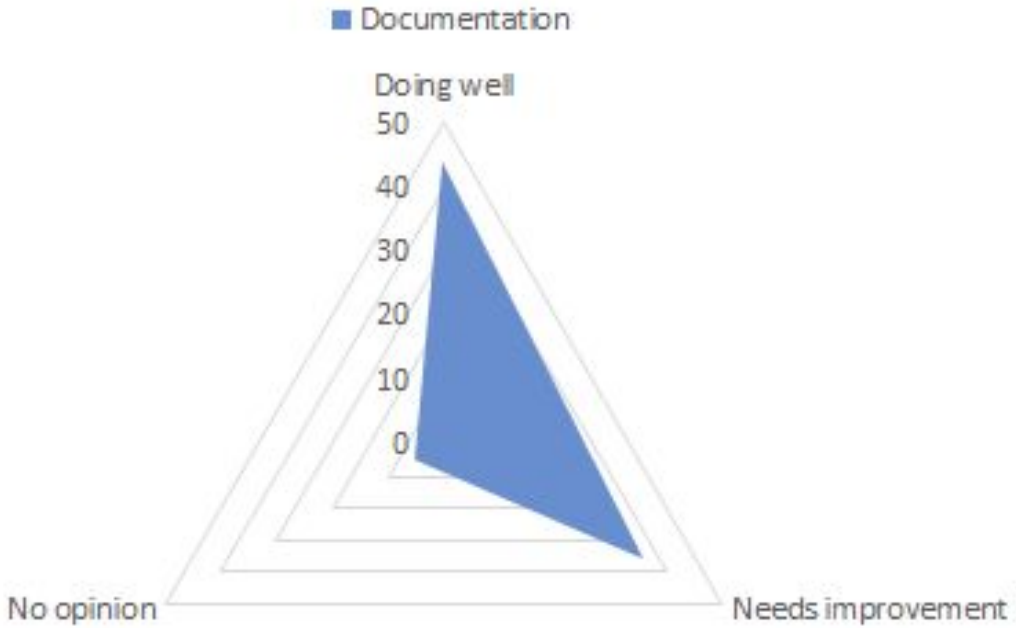
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



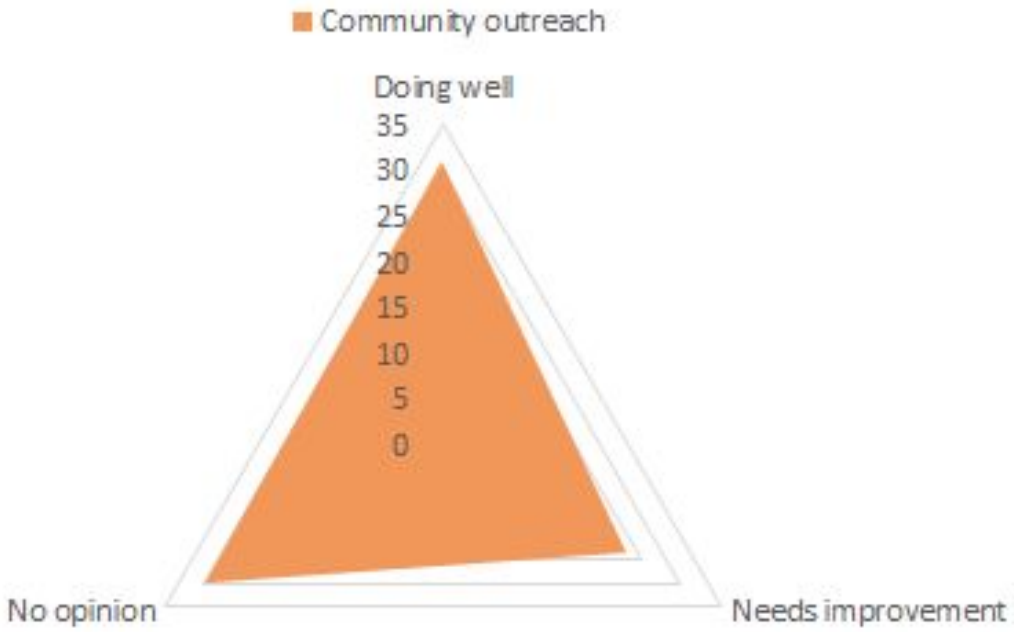
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



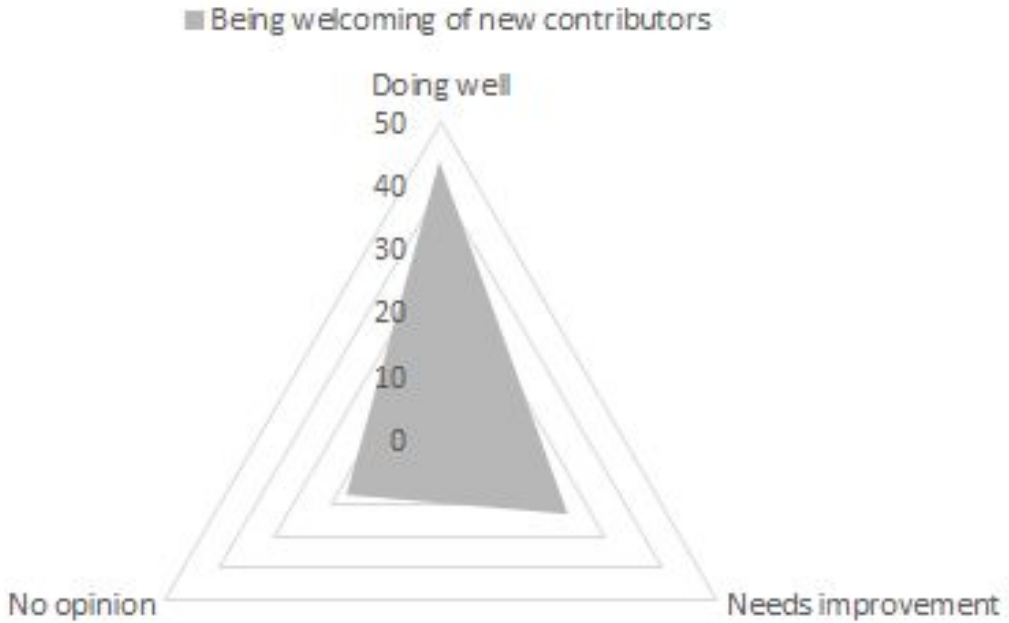
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



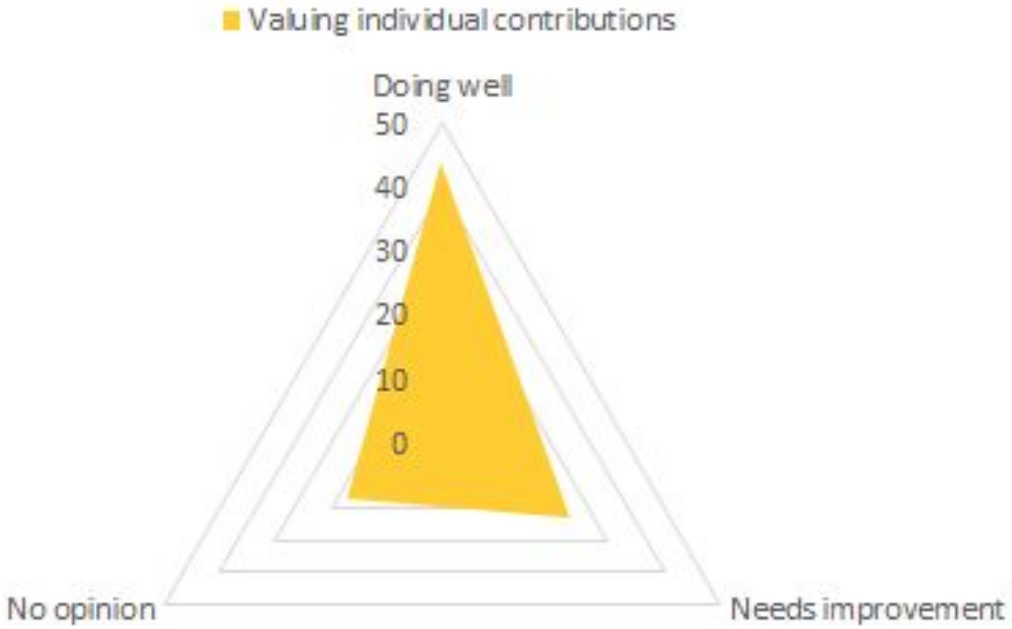
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement

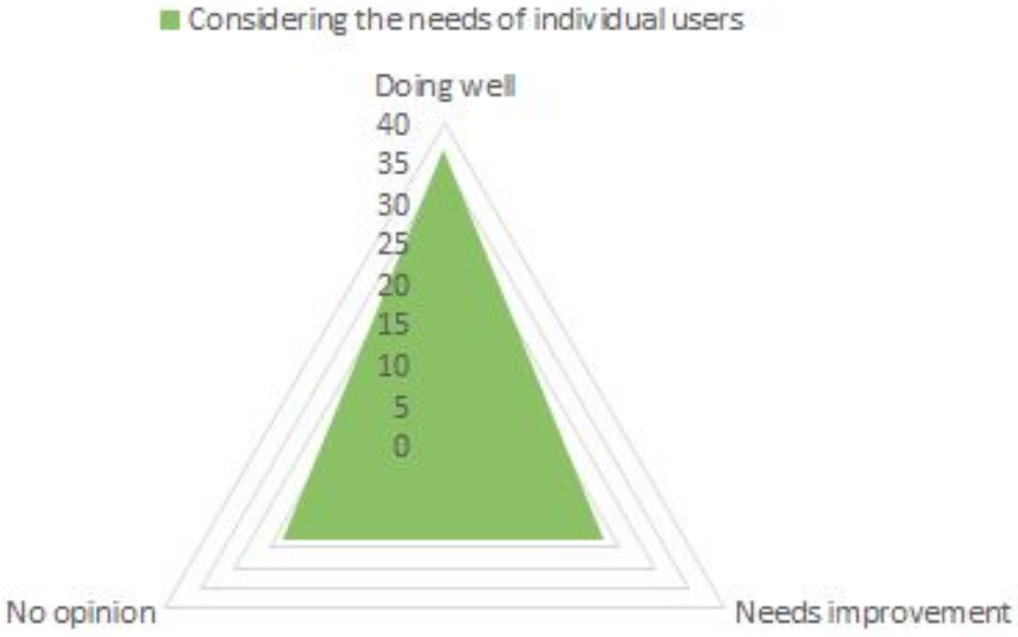


For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement

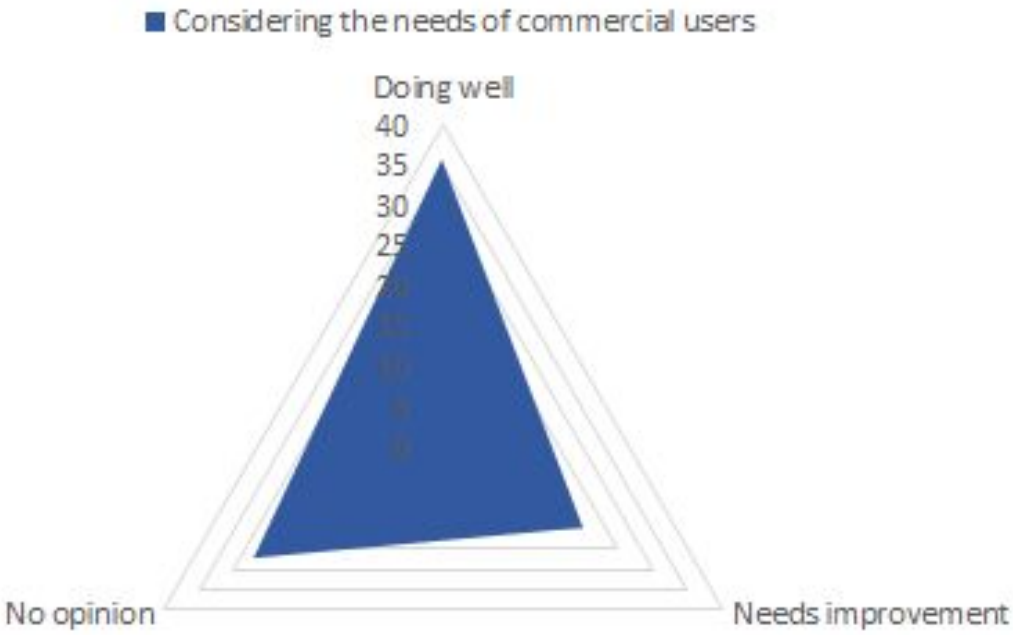


For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement

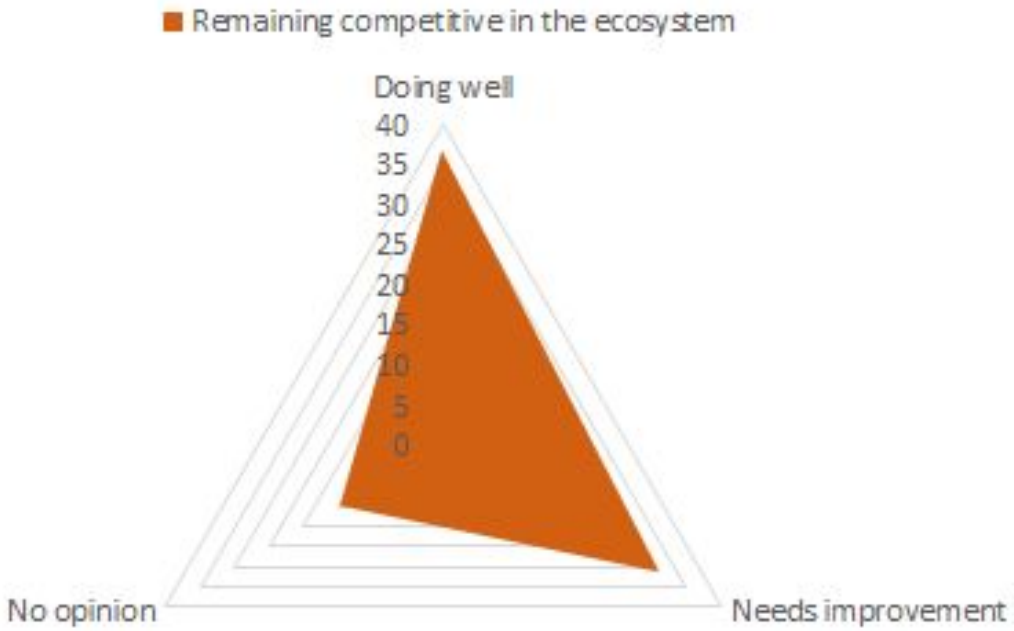
For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



For each of the following areas, indicate whether you think the Node.js project is doing well or needs improvement



Additional areas of improvement

- "It’s very intimidating and hard to contribute"
- "Technical expertise on macOS and Windows platforms."
- "Heated disputes among collaborators, they can sometimes give a hostile impression of the project. It deters me from sharing an opinion on specific PRs. Not getting involved and sharing no view on PRs that involve heated disputes is, admittedly, the easier option."
- "Velocity"
- "Focus on tech, not all this soft people/process stuff. As a semi-outsider I'd prefer an earnest attempt at meritocracy over all the "welcoming" talk that feels loaded with irony and cynicism."
- "We need more mentors!"
- "Explicitly document the process to become a core collaborator, to avoid individual bias."
- "Backwards compatibility, taken to the extreme, leads to software ossification. This ultimately leads to revolution where wholesale changes are adopted. Deno may very well end up doing that in the near term. If not, it is only a matter of time."
- "It is clear that much of the value of Node.js is in V8 itself and the module ecosystem. Deno adopts the former and is taking a calculated risk on the latter that may ultimately determine its fate. In addition to this, is is clear that many of the Node.js interfaces would have been designed differently today if language and web standards that later emerged were already in place. An obvious example is Promises... This rethinking also means that Deno can have a smaller core. If nothing else, they don't need two versions of the fs interface. The path to a smaller core may require taking one step back before taking two steps forwards... One of the most popular npm modules, request, was recently deprecated. Eleven years is a good lifespan, particularly in the context of the rapid evolution of the JavaScript ecosystem in that time. What do we want the, for example, fs interface to look like ten years from now? How do we build a plan to get us from here to there?"

Additional areas of improvement (continued)

- "Diversity in leadership"
- "At each release of Node.js, I see people asking for websockets in core. I don't have strong opinions but we have a hard time knowing what the users want. I'd be willing to orient my contributions toward features chosen through community outreach. But I am terrified the "small core" philosophy will shoot them down."
- "Native add-ons"
- "Benchmark data is difficult to see. ex: https://benchmarking.nodejs.org/charts/start_stop.png. It would be better to remove white border. And adding transparency might be good, or, smaller point might be good."
- "Core contributor retention."

Are there specific ways the project could make contributing easier for you?

- "Mentors"
- "I totally need help with i18n effort, because current contributors is not so active as I expect"
- "Building core and the *nix tools"
- "Less C++."
- "Clearer boundaries between subsystems / more isolated code."
- "Less ""internal only"" code."
- "A wish list or list of low-hanging fruit for features/refactors. Something to help those that have already completed their "Good First Issue" move on to their next contribution."
- "Enforce constructive collaboration. Minimize C++ code in codebase."
- "Contribution will be easier for Node.js lovers(JavaScript users) if it can be implemented in `lib`(JS) more. I think many of JavaScript users are not familiar with C++."
- "abandon consensus-seeking model, or do not allow non-TSC members to block additions...node should instead allow contributors to commit directly to core to address nitpicks. merge first, fix later. back changes out if there’s a disagreement."
- "More triaging of the `feature-request` label. It's hard to find feature requests that I can actually act on. The requests are often too vague, too complicated, or it's unclear if they're even something contributors would want done."
- "Use normal PR workflow, not this weird/scary push to master style. Also no idea why releases are so "manual" with what appears to be lots of "shit work.""
- "A more clear roadmap or wishlist to work towards"

Are there specific ways the project could make contributing easier for you? (continued)

- "More docs/comment on C++ code"
- "Ditch the review process. See next answer."
- "Maybe we could use an "experimental" fork, with less stability/perf but used to explore new features faster."
- "A quick review of the PR. Sometimes, I have to wait for long time (I do understand people are busy). After long time, either it became stale or we lose the interest on the PR. Again, it happened twice to me, but still we can improve on it."
- "Reduce some of the historical barriers that were introduced and are now followed for the sake of following them rather than because they serve a meaningful purpose."
- "Mostly when hitting issues with one or 2 contributors from IBM, it is easier to walk away from issues that continue to push for changes"
- "Documentation changes that could be made via direct edit on the web UI (like Mozilla does or did) could be one way. Merge via GitHub. Remove Python from all of Node. Anything that can be done to make local testing easier (eg: no makefile foo, perhaps even embrace an existing well-established testing framework)."
- "less tooling. Let us use github to land PRs."
- "Non-doc Good first issues and more transparent issue claiming process"
- "I am not good at English, sorry...I am newbie contributor...I feel very comfortable that the members are very kind and making quick gently responses...For first contribution, waiting time of merging is gnawed by anxiety, even though you make very quick ""approved"."

Are there specific ways the project could make contributing easier for you? (continued)

- "I think that there are many simple and trivial PRs in node/nodejs, for example, fix spell in documentation or source code comment. Such a case, it would be better to merge quickly, even if not important. It makes more contributors and will be involved longer, I guess. I expect that it would reduce members' burden, too."
- "making a board of orphan issues, need help usually has been taken by someone else"
- "If every collaborator could simply start a PR review comment with a positive word, before diving into the inevitable criticism. A renewed specific focus on being grateful to each other for our collective investment in the project. Making people feel valued."
- "Having more insight into the TSC and other committees project meetings, schedule, vision, etc."

What remains a blocker to becoming a more active contributor

- "Steep learning curve of testing"
- "Not enough activity from other contributors. I'm not asking to do stuff, I can do this itself, but I need PR reviews and discussions to fit with consensus process."
- "Consensus is too high a toll. Ideas like "small core" or "node.js re-imagined APIs" go against my desires."
- "The constant drama in the community"
- "More visibility about possible areas of contribution, at least in core"
- "Work commitments. I'm allowed to contribute on company time but am under pressure to justify the areas contributed to. For Working Groups, commitments to meetings late at night local time."
- "Time, complexity of changes"
- "Finding specific parts of core or new features to contribute to and knowing that the change would be appreciated."
- "Other commitments"
- "Consensus building is too hard, because too many overactive members are uninterested in engaging and simply block instead."
- "my own schedule"
- "No idea where to start or what to do. Seems only a certain group of people decide what should be done and they do it."
- "Time available"
- "A very small pool of very active contributors that push through their perspective by having a lot of time to argue."
- "onerous review process. nitpicky reviews. no clear support from other collaborators or TSC members on whether an addition is appropriate until it has become a PR (lack of a roadmap) very few “good first issue” issues"

What remains a blocker to becoming a more active contributor (continued)

- "Time & money"
- "An employer to support me doing it."
- "Native layer and legacy code, don't understanding V8"
- "Finding issues to work on"
- "The whole review process needs to be rethought. I make a change, update the documentation, make sure that it passes all the tests... and then I wait until other busy people get a chance to look at it. Instead of those people collaborating with me and making changes with skin in the game, they provide opinions on what could be done differently. I am then placed in a position where I need to defend my choices, or (and more commonly) simply acquiesce to what might be an inferior solution as that is the path of least resistance. Once I get sufficient preapprovals, I generally find that I need to rebase and start the process over... This entire process is demotivating. Most changes eventually go in. Git has the ability to revert changes that are found to be problematic. The process should optimize not only for preventing regressions, but also for making improvements and developer productivity... More fundamentally, I tend to develop incrementally. I break a task down into bit size pieces and implement each in one sitting. I take care that the system is never broken, but the new feature or bug fix may not be complete until the series is complete. In other open source projects, I may do one to three commits per day. In Node, I have to assume that each commit will take a week. That changes how I approach things: makes me consider larger (and less manageable) chunks, and working on other unrelated things while I wait for a review... Flip the script on reviews. Ask those that see better ways of doing things to either invest the time in making the change themselves, or put the burden on them to convince others rather than giving the the power to withhold approvals."

What remains a blocker to becoming a more active contributor (continued)

- "I am scared by the time it can take to have a PR merged if it is an important one."
- "I personally comment/review repl codebase. I always wanted to do the c++ side of it, but somehow the documentation or getting started with c++ is tricky I feel."
- "The weight of any contribution to nodejs/node that's not incredibly complex and the lack of tooling and automation that reduces the strain on humans."
- "Time as I do it on my own time."
- "Personal time and interests have shifted a bit since I became an engineering manager."
- "Time for open source."
- "Time to find a good issue."
- "Lack of non-doc good-first issues"
- "A better guide on how to run only parts of tests, since make test takes a long time. For fast iteration when working on a specific file."
- "my spare time is too short."
- "Finding manageable tasks"
- "time"
- "Free time"
- "clear understanding of what the project needs."

What remains a blocker to becoming a more active contributor (continued)

- "It's always a struggle, and I value peace of mind"
- "No real reward for contributing, apart from seeing contributions ship."
- "Financial limitations (time)."
- "Knowing what to work on"

What would you remove from the project?

- "Http2"
- "workers"
- "The old (callback) APIs, streams (redo I/O interfaces) and error handling"
- "Domains, unanimous consensus, the REPL module."
- "The community drama, reduce the wait time for prs to land"
- "The constant drama. The "cliquey-ness" of the project - If 2 or 3 people dont like you, you are not welcome to the group. One person seemingly has the ability to block things even if everyone else thinks its the best way forward."
- "the installers, package manager (npm), es module's use of package.json (as opposed to leaving package.json to the ecosystem)."
- "cluster, legacy url parser, streams"
- "Modules"
- "The .mjs extension."
- "Bureaucracy"
- "CommonJS, callbacks, separate http(s)2 APIs"
- "Semi-colons, LTS (the confusing name, not the concept), Readme credits (collaborators/members) section"
- "The headers to support any native module interface other than N-API, the cluster module, and just about everything in the vm module."
- "npm"
- "v8 flags"
- "senior contributors public harassing junior contributors work."

What would you remove from the project? (continued)

- "CJS, Callbacks, Buffer"
- "I would lower the consensus bar."
- "Normalize our structures. Everything is a WG, and centralize all governance information into a single `nodejs/governance` repository rather than having it split across the org in a way that's completely undiscoverable."
- "Our docs. Nuke them. Replace them with a *system* that's maintainable and scalable rather than an amalgamation of 11 years of technical debt that has a non-trivial number of errors that confuse or complicate learning the platform."
- "Event emitters and streams. Don’t have a third. Promises all the way for internals and async iterators for the streams usecase."
- "node-gyp based build system, but not if replaced by a Ninja based replacement"
- "Streams, EventEmitter, callbacks (because Promises)"
- "domain, unhandledRejection not crashing."
- "keeping modules registry apart from Node.js governance"
- "Buffer, legacy url module"
- "Only EventTarget for now"
- "GYP"
- "domains, async_hooks, the "spirit" that performance is more important then correctness (at least if feels like it sometimes)"
- "Move network security related modules (ssl, tls, https, http2 and so on) to other repository"
- "Revamp all builtins as Promised-based ES modules with attached security policies"
- "Remove all high security access functions on the process global and move them to security-policy-managed imported builtins."
- "Extend / revamp the ""experimental policy"" system to something that can encompass a more general security policy system including per-context-security creation in future."

What would you remove from the project? (continued)

- "CJS => convert to full ESM"
- "callback patterns in all APIs => convert to promises"
- "GYP, lets use make, anything else"

What would you add to the project?

- "WebSocket core API"
- "Official musl builds"
- "Less white dudes"
- "How about adding a reward system for active members? For example, Gatsby.js send swags for contributors, but if we can’t also spend money for that, then at least we can create messages on Twitter with thanks"
- "Domains, CommonJS (one can dream)"
- "Consistent errors, consistent cancellation and possibly better integration for tooling like TypeScript"
- "Better package manager or improve npm to be on par with yarn on features and stability"
- "Ways to use web APIs, a written up / clear security model, TS compatible JSDoc types to core"
- "put the experimental modules flag back on 12, make it clearer how to get more involved in core development,"
- "a code-of-conduct that is fair to ALL parties and cant be abused to remove members one person doesnt like."
- "Promises compatibility for all async processes"
- "A more "standard" non-gyp based build system"
- "Ability to build stand-alone binaries with altered builtins and an embedded main script."
- "Ability to disable experimental warnings with a CLI flag."
- "A wish list or wanted features list (perhaps with a 'mentor available' tag)."
- "websockets, fetch"
- "A clear set of target usecases - Node was a serverside javascript runtime - this has changed but there is ambiguity in what that means."

What would you add to the project? (continued)

- "A new namespace like @nodejs where we can put redesigned APIs that follow modern patterns, like @nodejs/fs that is what's now fs.promises."
- "Rust instead of C++, fetch, supports for iOS/Android"
- "Fetch, EventTarget, official instrumentation hooks"
- "QUIC support."
- "Fewer inconsistencies in the http(s)/2 APIs…"
- "Maybe native `fetch` support? (I know... there are a lot of Web Standard questions that would need to be resolved. This is a wishlist.)"
- "WebRTC/codec support, capability based security model, comprehensive Isolate support (e.g. Lambda, Edge Compute, Workers)"
- "WebSockets, a "blessed" generic resource/connection pool, a built-in test/benchmark system similar to Rust"
- "top level await, fully promisfied interface, typescript"
- "ES modules, and "friendly" typescript support even if I have no idea what that means."
- "Continuous local storage, BOB (new streams)"
- "Release automation: automated back-porting and releases. We have a non-trivial number of errors in releases that are a good indicator of our failure to make it simple (I would absolutely never blame the Releasers on this - the process is massive)."
- "Rewrite meeting tooling: currently being done in pkgjs, but really invest in making something that drives so much work in the project *really* solid."
- "Org-wide standardization: please glob can we standardize various automation tooling and make it accessible to everyone."
- "TypeScript, promisify absolutely everything asynchronous by default (not via /promises indirection)"

What would you add to the project? (continued)

- "CRASH ON UNHANDLED REJECTIONS. CRASH ON UNHANDLED REJECTIONS. CRASH ON UNHANDLED REJECTIONS."
- "Node.js funded modules registry, removing ESM flag in Node v12"
- "Monitoring"
- "Fully promisified api's as default."
- "Streamline acceptance of new members."
- "replace async_hooks by a more user friendly, higher level API add some more utilities now spread in various userland modules more source code documentation"
- "Set C++17 or C++20 to the minimum standard of v8 and nodejs."
- "auto track of issues that is not complete because of lack of acceptance (not leaving PR on air, reject or accept)."
- "Fetch, fetch, fetch"
- "make or anything else for building C++ addons"
- "native ICMP"

Additional Comments

"I get a consensus-seeking model. it makes sense for standards bodies. Node is not a standards body and needn’t assume that overhead.

if certain individuals have been noted to repeatedly refuse to compromise, these individuals should be removed from a decision-making position. I don’t have names, but I can make a few guesses on repeat offenders. even if it isn’t a CoC violation, there’s no reason why the TSC couldn’t decide (probably by vote?) that certain contributors should be disinvited. if that fractures leadership, so be it—new leaders will come along. no single person’s technical ability or knowledge should be allow to make contributing worse for many other people!

avoiding the issue is like taking on debt; spare the long-term conflict of trying to work with these people, and instead take the conflict and discomfort now."

Additional Comments

"What honestly inspires me when it comes to the future of the JS ecosystem, is Deno. It does a lot of things well, unsurprisingly perhaps, as it started with a clean slate. I'm incredibly aware of how hard it is to adopt some of its popular/best practices in an existing system like Node. Having TypeScript out of the box for example, would be of huge benefit to a very significant part of our userbase. This goes well beyond ""I know no longer have to put TS in my project myself"". It would probably push a shift for libraries towards TypeScript, where they seem to be headed anyway, and a move away from DefinitelyTyped (which I believe is a wart of our ecosystem that we need to get rid off)."

Additional Comments

"I love you all, keep up the excellent work!"

"Thank you very much for great work. I respect you and the community so much!"

"Node.js is amazing. I've converted many companies/company projects to it and done my best to convince them to contribute back. If not directly to Node.js then to the open source community."

Questions we should ask next time?

- "Why did you contribute to Node.js?"
- "The meetings... I wanted to join meetings, but its always on my night time. It would be great, if we can keep at least a month meeting on [my local time] it would help :)"
- "the modules team has been a problematic interaction with the Node.js project for most people. This needs to be addressed."
- "How to improve the process contribute to Node.js will be good question."
- "How did you get involved in Node.js?"
- "Perhaps some more proactive / open questions around how to overcome any issues as opposed to purely identifying them. There are some tough problems around ensuring the health of the project and community and getting those suggestions for figuring out how to solve them would be beneficial."

Thanks to core contributors for helping...

* **Anna Henningsen** - Special callout for getting the most thanks!

* Ben Michel
* Ben Noordhuis
* Gabriel Schulhof
* Guy Bedford
* James Snell
* Joyee Cheung
* Matteo Collina
* Michael Dawson
* Myles Borins
* Rich Trott
* Rod Vagg
* Ruben Bridgewater
* Sam Roberts
* Trevor Norris
* Yosuke Furukawa
* Zeke Sikelianos
* Anatoli Papirovski
* Brian White
* Fedor Indutny
* Glenn Hinks
* Gus Caplan
* Jeremiah Senkpiel
* Matheus Marchini
* Michaël Zasso
* Rebecca Turner
* Refael Ackermann (רפאל פלחי)
* Richard Lau
* Roman Reiss
* Ryan Dahl
* Shelley Vohr