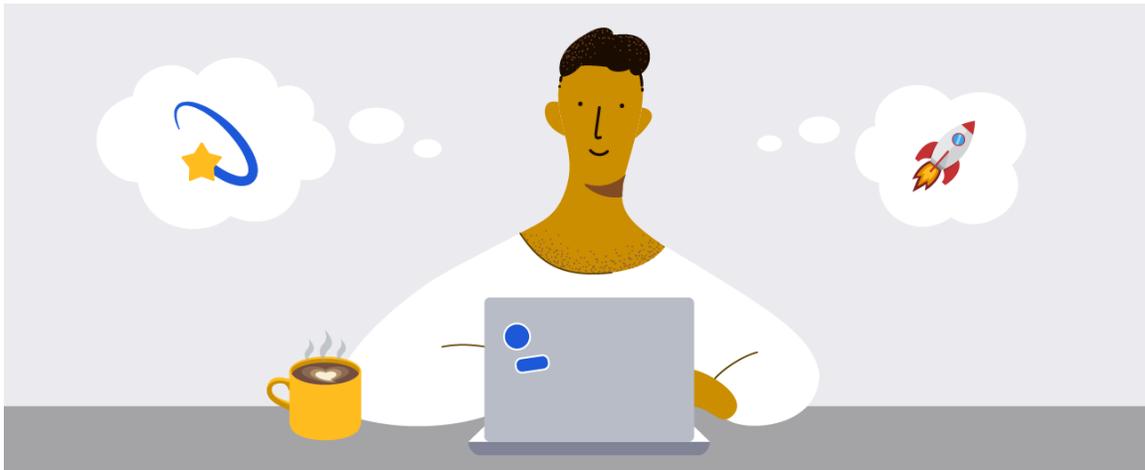




 Mattermost

How to Build Your Open Source Productivity Stack

Learn why you should make open source software part of your organization's productivity stack, and how to get started today.



What does your team's dream productivity stack look like?

Developer teams want tools that are enjoyable to use, are flexible enough to adapt to the unique tools and processes in their workflow, and do what they're supposed to do. But it's not uncommon for the tools that teams use to be a poor fit for the actual work that they do.

The productivity tools we use impact the success of our businesses. Great tools reduce friction, improve communication, and increase our ability to produce excellent work effectively. On the flip side, tools that are a poor fit can hinder productivity, reduce job satisfaction, and limit the speed and efficiency with which we can collaborate.

If you're reading this guide, you probably already know that open source tools can be an essential part of your team's stack. In fact, your team is probably already using a lot of open source tools as part of their development workflow. Either way, there are some advantages to intentionally choosing open source tools for your organization.

With a rapidly expanding field of enterprise-ready open source tools on the market, there's no reason your organization — large or small — can't take advantage of these powerful and flexible tools. Thanks to this proliferation of open source tools, this is the first time you can open source your whole stack and function at an enterprise level.

As Paul Cormier, president and CEO of Red Hat, wrote in [The State of Enterprise Open Source 2022](#):

“Some technology persists for decades if not longer, and the decisions IT leaders are making today will impact their organizations’ nimbleness and market response down the road, whether it’s in two years or 20. As new infrastructure is being built out, you can’t leave behind existing systems and tools. You need products and services that work with them. That’s the value of open source.”

In this guide, we’ll explore the advantages of using open source tools as well as the challenges and misconceptions that you may face in getting buy-in and implementation. We’ll also cover how to choose open source tools that will be a great fit for your organization so that you can evaluate the solutions your team wants to use — no matter what’s in their stack.

What are the Advantages of Open Source Tools?	4
Challenges and Misconceptions of Open Source Tools	7
How to Choose Tools for Your Open Source Stack	10
Why Developer Teams Choose Mattermost	13

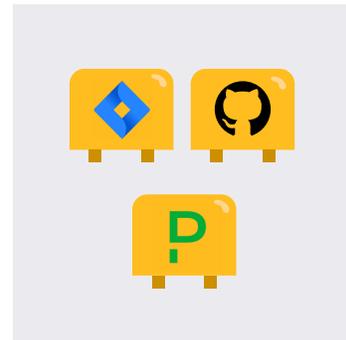


What Are the Advantages of Open Source Tools?

Flexible software integration options

Developer organizations are facing a crisis of fragmentation right now; there are more tools available than ever to help them collaborate, code, and maintain projects. But understanding, vetting, and integrating these tools is a daunting task that can require significant time.

Thanks to their open code bases and developer-accessible APIs, open source tools are often far more integration-friendly than proprietary tools. They may come with ready-to-use integrations with the most popular tools in your stack and, if not, they can be configured for integration with almost anything your team needs.



Because open source tools are often built using the same underlying principles (and sometimes even by overlapping members of the open source community), they tend to play nice with each other. This baked-in compatibility makes them even better when you open source multiple tools in your productivity stack.

Total control and customization of your software

One of the biggest reasons that productivity and collaboration tools fail is because they're a poor fit for what the team needs. Developer stacks are unique to each organization, and

developers prefer tools that fit the way they work rather than having to adapt their workflow to fit the limitations of a tool.

Productivity tools are hugely popular, but many of them [fail to make an impact](#) in the long run. Open source tools can be customized to your organization's unique workflows, improving their fit, increasing adoption, and enabling your team to take full advantage of these tools.



Improved security oversight

According to the Red Hat [State of Enterprise Open Source](#) survey, 89% of IT leaders believe enterprise open source products are as secure or more secure than proprietary software. Teams with the highest security standards often choose open source software because of their ability to audit the source code for themselves, great documentation, and security patches.



In fact, high-security organizations like the [U.S. Air Force](#) are adopting more open source solutions than ever because of the transparency and flexibility they offer:

“As the DoD, we are not a very trusting enterprise by design,” said Master Sgt. Matthew Huston, the team’s chief of enterprise services, during an interview with Inside Defense. “What we do at Platform One is buy down our risk with vulnerability scans upfront as well as constant monitoring while the tools are in use. This allows us to move faster across more tools knowing that there is someone/something always watching.”

Crowdsourced innovation from the open source community

Open source tools are driven by community support, which influences the way they grow and evolve. Unlike proprietary solutions, which are often hemmed in by an internal product roadmap and priorities, new features and functionality for open source software can come from anywhere.

A healthy, robust open source project might have innovative features, bug fixes, and integrations coming from developers who are actually using and relying on that software every day. And because anyone can contribute, these fixes and innovations often come faster than you might see with proprietary tools.



There's another benefit to using innovative open source tools; not only will you be using tools built by some of the best developers in the world, you'll also be better able to attract developer talent by using and investing in cutting-edge technologies. If you've significantly improved the solution, that serves as an opportunity for your developer team to contribute back to the open source developer community (if it's feasible for your organization to do so). These contributions help push the boundaries of technology and help the industry as a whole innovate. Put simply, using and contributing to open source tools can position your developer team as thought leaders.



Challenges and Misconceptions of Open Source Tools

“Open source requires too much maintenance.”

A survey from Tideline on [open source usage by professional teams](#) found that about a quarter of the time developers teams spend on code maintenance is specific to open source tools. Much of the maintenance work amounts to updating to new versions or adapting to bugs as required.

Because they can be customized to fit your team’s specific needs, open source tools can absorb as much work as you want to put into them. But just because you can fork and customize an open source solution to your heart’s content doesn’t mean that it inherently needs a lot of maintenance. The maintenance required is defined by the amount of customization you put into it.

Using open source tools as they are envisioned by their developer communities will keep you in alignment with the tool and its ongoing maintenance. For teams that need to further minimize their maintenance time, some enterprise open source solutions offer cloud versions of their platforms.

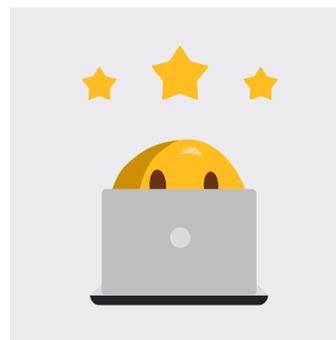
And there’s an advantage to the maintenance that some teams may do on their open source tools. Contributing improvements back to the community reduces the amount of custom maintenance you’ll need to do in the long term because open source maintainers can help take on some of the load.



“Open source software isn’t user-friendly.”

Historically, designers and UX specialists have been underrepresented in the open source community. But that’s beginning to change, and the landscape of open source tools—especially for enterprise users—has evolved.

“A distinguishing feature of open source software versus proprietary software is that open source software tends to be used by a diverse community of users with different priorities, needs, use cases,” writes [Greg Myers](#), support engineer at GitLab. “I personally feel the more diverse and inclusive that community is, the better the end product is.”

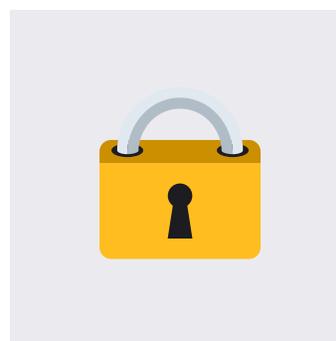


Many open source tools are now much more user-friendly than earlier versions and can be used not just by your developers, but by non-technical members of the team, too. As enterprise open source solutions have gained traction—with all the design, usability, customer support resources that they can offer—organizations have gained more access to open source alternatives that aren’t just functional, but enjoyable to use.

“Open source software is not secure.”

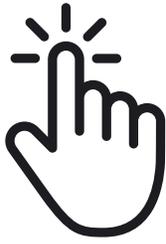
Whether any one piece of open source software is inherently more or less secure is hard to determine; but what open source does offer is visibility, which has made all the difference to teams concerned with security.

The thinking around the security of open source software in the last 10 years has completely flipped; [89% of IT leaders](#) believe enterprise open source solutions are just as sure—if not more secure—than proprietary solutions. The reason behind this shift? Many teams now recognize that access to source code allows their in-house security teams to fully vet the code more easily than they can with proprietary software. They’re also able to adapt the source code to align with



any compliance or security requirements that may be unique to their organization.

Many organizations are putting their money where their mouth is on this front; some, like Spotify, have [created their own funds](#) to support open source projects. Others—including tech giants like Amazon, Google, and Microsoft—[invest in organizations](#) like the Open Source Security Foundation (OpenSSF) to help open source projects improve quality and security for all users.



How to Choose Tools for Your Open Source Stack

There's no one-size-fits-all open source solution that will work for every team. Your tolerance for experimental features vs. stability, price point, compatibility, scalability, and team preferences all factor into how you choose your tools. In addition to your selection criteria for any tool your team would use, there are a few additional considerations for open source productivity tools to take into account:

Support: Resources to help you set up and manage your open source tools

If navigating the configuration and maintenance of an open source solution is daunting, don't worry—you might not have to go it alone.

Enterprise open source or COSS solutions may offer support packages that rival traditional SaaS support options. Not every team needs support, but if yours does, look for open source organizations that offer the level and kind of support that will help ensure successful use of the tool.

What to look for:

- Robust product documentation
- Clear introductory materials and tutorials across supported platforms and deployments
- Advanced help articles that provide a breadth of troubleshooting information

- Dedicated customer success resources
- Active community support channels

Scalability: Infrastructure and features that allow you to scale tools with your team

When devs adopt tools for themselves, scalability doesn't matter; when you're rolling it out to an entire organization or team, it does. Look for open source tools that are designed to grow with your organization; adoption can be challenging, and the bar for switching tools at scale is high. One way to get a sense of scalability is to look for large organizations using the project. If they can successfully use an open source project at scale, there's a good chance you can, too!

What to look for:

- Adoption by large organizations
- User and data limitations
- Deployment limitations (how well will this tool integrate with your stack at scale)
- The cost of the platform at scale (if any)

Learn more about open source scalability

Scalability is about more than technical considerations. Everything from security to usability can impact how your organization uses a software solution at scale. Learn how one Fortune 50 organization rolled out an open source collaboration platform to 34,000 users. [Watch now »](#)



Usability: Functionality and guides to improve user adoption

Productivity tools that aren't usable aren't likely to be adopted; vet your tools not just on what they promise, but how they perform day-to-day. If possible, get a few members of your team to take new tools for a trial run and get a feel for how it fits into their workflow.

Make sure that the tool you're evaluating has the resources you'll need to get you started, from onboarding guides to in-depth documentation. Those onboarding guides should give you a better idea of how quickly and easily you'll be able to set up the tool and start seeing value from it, so take the time to review those guides as part of your evaluation process.

What to look for:

- End-user features designed to improve workflows
- Resources for end-users to implement their own tooling or workflows
- Integrations with other tools in your team's stack
- Ease of customization

Active community: A diverse, engaged open source community to keep the software moving forward

One of the key benefits of open source is harnessing the power of a broad, diverse community. While support from the primary organization is important, contributors are the lifeblood of any healthy open source tool, contributing innovative features, essential bug fixes, and improving documentation for everyone. To get the most out of your open source tools, look for those with an active, engaged community.

What to look for:

- An active repo with frequent contributions
- A large number of external contributors (people from outside the primary organization)
- Trustworthy leadership from the organization
- Rapid response to contributors with follow-ups to find a resolution

Why Developer Teams Choose Mattermost

Whether you're an open source veteran or just starting to experiment with open source tools, Mattermost is the perfect open source productivity platform to add to your team's collaboration stack.



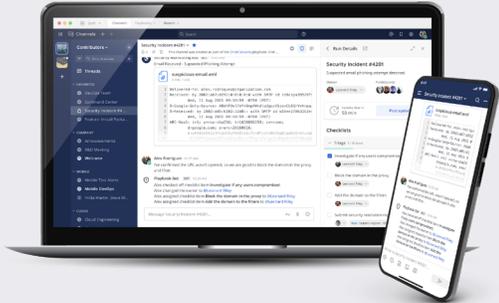
Designed for R&D Workflows: Mattermost is built for developers teams with features that help you collaborate when coding, sprint planning, deploying, and during outages and support escalations. Easy to add integrations, slash commands, API & automation, all built from open source tools that can be reconfigured and customized.



Supported by a Global Community: Mattermost has been open from the start, and is supported by a global community of over 4,000 open source contributors.

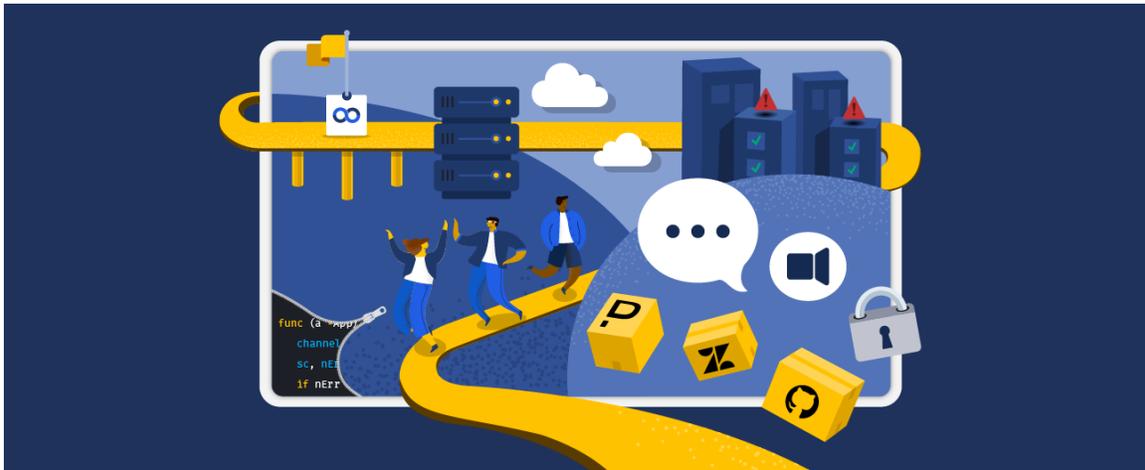


Created for Secure Collaboration: Deployable in both on-prem (or air-gapped) environments and secure cloud — we are the only solution that gives you complete control over your data.



Next Steps in Open Source Collaboration

Want to try out Mattermost for yourself? [Deploy your own Mattermost server](#) in a matter of minutes and get started now.



About Mattermost

Hundreds of thousands of developers around the globe trust Mattermost to increase their productivity by bringing together team communication, task and project management, and workflow orchestration into a unified platform for agile software development.

Founded in 2016, Mattermost's open source platform powers over 800,000 workspaces worldwide with the support of over 4,000 contributors from across the developer community. The company serves over 800 customers, including Samsung, Nasdaq, SAP, European Parliament, and the United States Air Force, and is backed by world-class investors, including Redpoint, YC Continuity, Battery Ventures, and S28 Capital. To learn more, visit www.mattermost.com.



Mattermost

[Mattermost.com](https://mattermost.com)

