

Onboarding Technical Talent: Setting up New Hires for Long-term Success



Everyone's been there at some point or another: You're new on the job, sitting in a meeting with a bunch of people you *sort of* know, and they're talking about a project you're *kind of* familiar with. But as they dig into the latest project's details, they might as well be speaking Ancient Egyptian.

While the interview process can feel like the biggest hurdle to bringing a new member of the team into the fold, getting a new hire up to speed often takes months, and — if mishandled — can leave your newest team member feeling like a perpetual outsider. This is even more important for remote team members who don't have the advantage of being able to connect with people in person in low-friction, unplanned ways.

What does a successful technical onboarding look like?

On average, [it takes about six months](#) for new hires to become productive members of the team. But that doesn't mean you can't beat the curve.

Equipping the newest member of your team with a game plan to help them onboard effectively will help ensure that their first few days, weeks, and months are successful. They might still feel like they're reading hieroglyphics, but at least they'll have the tools needed to decode them.



Rolling Up Your Sleeves

Why it's important

Developers want to build things. The first few days and weeks can be a lot of training sessions, paperwork, and getting-up-to-speed work that doesn't feel productive.

Giving them the opportunity to get their hands on the code and start contributing is one of the best ways to make them feel welcome out of the gate.

Putting it into practice

- Set small goals that ramp up
- Give them time and permission to start working on things
- Get them involved in processes and workflows early
- Provide documentation so they can self-start



Connecting with the Team

Why it's important

Your candidate met a handful of people during their interviews. Now it's time to really make them feel like part of the group — not just a guest.

Create space for them to get to know their new team both 1-on-1s and in larger team settings, especially in less structured forums than existing meetings. If you don't already have regular team lunches, happy hours, or game nights, now might be a great time to start a new tradition!

“Communities of practice play a huge role in assimilating new engineers. Connecting new hires to relevant communities of practice helps to quickly create a sense of belonging and share knowledge.”

— Gartner, Inc., “How to Create a Frictionless Onboarding Experience for Software Engineers,” Manjunath Bhat & Thomas Murphy, 17 February 2021

Putting it into practice

- ☑ Set up 1-on-1s with peers, managers, and mentors
- ☑ Have hiring “buddies” for the first few weeks
- ☑ Hold dedicated “welcome” meetings or sections during existing meetings
- ☑ Create regular events for the team to socialize



Knowing Where to Find “Stuff”

Why it’s important

Visibility into processes, workflows, and historical information makes all the difference between feeling like the “new kid” and an old pro. After all, most folks don’t want to spend their time endlessly searching for information or guessing on the right way to get things done.

Make sure that new members of the team have everything they need to work productively starting on day one.

Putting it into practice

- ☑ Provide a clear onboarding roadmap with links to key information
- ☑ Prioritize keeping processes documented and easy to find
- ☑ Encourage communication in public channels rather than DMs

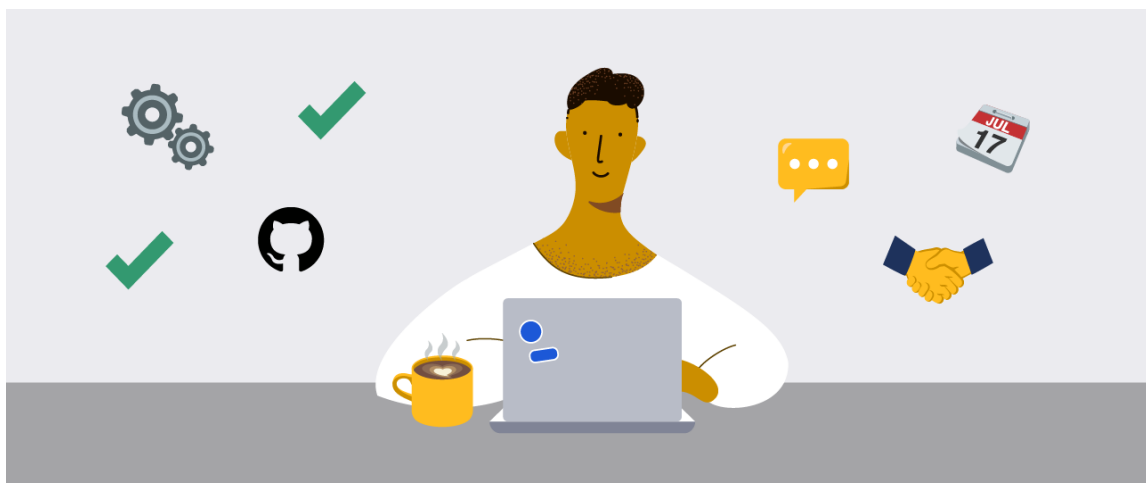
Technical Onboarding Playbook

No matter what role your new hire is onboarding for, there are a lot of moving parts involved in bringing a new person onto your team. Luckily, a clearly documented, well-defined onboarding process can make that transition a little less overwhelming for everyone.

Every organization's onboarding processes will be a little different, but there's a common pathway for success for most technical teams. Check out our Technical Onboarding Playbook — based on [the one we use here at Mattermost](#) to onboard new members of the Engineering team — to learn what the first four months can look like for a new developer.

Week 1: Focus on environment setup and introductions

- Set up equipment and environment
 - Get your hardware and set up dev environments, clone the repo, and set up accounts. Expense any equipment needed and finish HR setup.*
- Meet the team
 - Meet with a different team member for 30 minutes each day*
- Meet with your dev lead
 - Meet for at least 10–15 minutes each day*
- Attend weekly team/company meetings to get the feeling for them
 - Sprint planning, company all-hands, dev meetings, dev hangouts, etc.*
- Join important messaging channels and project boards
- Read and understand the org's core values
- Work on 1–3 small tickets to get used to dev processes
- Introduction at the R&D meeting



Week 2: Focus on digesting information dump

- Continue getting used to processes and workflows
 - Attend some spec/design/technical meetings; your dev lead should send you invites to these meetings*
 - Do some code reviews; your dev lead should assign you to some*
- Introduction meetings with a different team member each day
- Mentor for the week is a senior teammate
- Meet with dev lead 2–3 times in the week and review company values
- Work on 1–3 small tickets
- Introduction at the company all-hands

Weeks 3–4: Focus on solidifying role in the team

- Start taking a more active role as a member of your dev team
- Begin to participate in some spec/design/technical meetings
 - Provide feedback/ask questions, have an active role in the meetings*
- Mentor for each week is a different teammate
- Start regular 1-on-1s with dev lead
- Work on some medium-sized tickets



Weeks 5–8 (Month 2): Work on your first project as dev owner

- Begin work on your first relatively large task and be the dev owner of it
- Act as the dev owner in the spec/design/technical meetings
- Write up a brief technical specification if necessary
- Work on small/medium tickets each sprint alongside the larger task

Weeks 9–11 (Month 3): Work on more and/or larger projects as dev owner

- Take what you learned being dev owner on your first project and improve upon it
- Act as dev owner and write technical specs for more and larger projects
- Work on small/medium tickets to fill in gaps
- Complete deliverables in a more timely fashion
- Attend a “Tech Moonshots” developer meeting

Week 12: Informal performance evaluation

- Dev lead will give you your performance evaluation
- A self-survey will be sent to you
- Surveys will be sent out to your peers and you are:
 - Evaluated on what’s being done well, what could be improved*
 - Evaluated on core values*

Weeks 13–16 (Month 4): Act on your performance evaluation and focus on broader engagement

- Begin acting on any feedback given during the performance evaluation
- Start taking a more active role in the community and customer engagement
 - Proactively answer questions in channels*
 - Create Help Wanted tickets*

THE BOTTOM LINE

Lay the groundwork for success

Your newest team member wants to start making meaningful contributions right away, and it's your job to give them everything they need to hit the ground running. The right tools and onboarding processes are essential to getting people up to speed and working quickly, which helps set the stage for a happier, more productive team in the long run.



Create a roadmap to success. Your new hire has a lot of catching up to do. Make it easy for them to navigate your organization with clear documentation, checklists, and tutorials.



Prioritize relationship-building (especially for remote teams!). The first few months can be an information overload, and finding a connection with team members can mean the difference between sinking and swimming. Plug your new hire into the existing team both as a group and one-on-one.



Establish check-ins early on. Your new hire might be eager to dig in. But that doesn't mean you should let them tackle every challenge on their own. Check in with them on a regular basis to make sure they're on track.



Learn more about building your technical team

Want to learn more about building and retaining a high-performing technical team? Fostering a highly-effective development team is an ongoing journey that starts and scales with an amazing team culture. Be sure to read the first part of this series, **Building a Developer-Centric Culture**, to learn more about the qualities that developers look for in the organizations they join.

[Read More](#)