How managers use great developer experience (DX) to empower top-performing teams

stackoverflow
FOR TEAMS

Introduction

Most companies place a high value on the customer experience (CX), or how customers experience interactions with your brand. Throughout the customer lifecycle, the goal is to provide a seamless, intuitive experience that meets customers' needs at every touchpoint.

But without a great development team in place to build features and products in response to market demand and to quickly fix bugs, your customer experience will quickly begin to suffer.

When you consider each developer another kind of "customer," you can provide the kind of engaging, stimulating, and rewarding work environment that empowers top-performing teams.

That's why it's important to place just as much emphasis on the developer experience (DX), which refers to "how developers think about, feel about, and value their work."

Let's consider what goes into the developer experience and what your company can do to create a seamless work experience that helps you source and retain top tech talent while empowering developers to continually upskill, collaborate, and innovate.

What goes into the developer experience?

To build a great developer experience, your company must be empathetic towards what developers need to do their jobs efficiently and remain engaged in the process. That means it's essential to take a holistic view of your developer tech stack and workflows to generate insights into what's working and what's not—and come up with strategies to eliminate blockers that prevent developers from meeting their goals.

Crucial elements of a good DX include:

- **A seamless onboarding experience.** When your developers join the team, how quickly can they start being productive? Your onboarding process should be streamlined enough to ensure that developers have access and permissions to all the tools they need to use on their first day, so they can start exploring and learning, rather than waiting around for approvals.

- **The ability to deploy code quickly.** In a good developer experience, your development team should be able to move from completed code to production in one day.
  The deployment process should be simple and streamlined, with the ability to create phased feature roll-outs based on set criteria.

- **Access to the right tool set.** The tools and technologies that your developers have access to should be driven by their needs and interests—not by the IT budget. When your team members are advocating for updates to your tech stack, listen to their reasoning and let them drive change.

- **Access to information and collaboration.** Developers can overcome productivity blockers by getting access to information when they need it and being able to share their own insights with their peers in an asynchronous format that doesn't pull them away from their other work.

- **Smooth interoperability.** Developer experience includes all of the other elements that go into product development: maintaining code, testing, security, addressing incidents, and more. Your tech stack should ensure that developers have a smooth workflow with clear interoperability of tools to enhance productivity at every stage.

By paying attention to what your developers are looking for in their own professional development, you'll be able to offer them a rewarding environment where they can continually learn on the job and become even more valuable to your organization.

Prioritize developer experience to attract and retain top talent

Specialized tech talent is hard to recruit and even harder to keep. According to the Bureau of Labor Statistics, software developers are one of the top four in-demand professions in the United States, with around 200,000 positions open every year—and only a fraction of candidates are available with the right skill sets.

If and when you manage to find a candidate with the right skills, it can be even more difficult to keep them long-term. A survey from TalentLMS and Workable found that 72% of tech workers are considering leaving their jobs in the next year, with the most common reason (40%) being limited career progression opportunities.

In order to both attract and retain great tech talent, it's essential to build a strong developer experience that provides your team with the tools and technologies that empower them to gain new skills on the job: 91% of employees say they'd like more learning and development opportunities.

91%
of employees say they'd like more learning and development opportunities

Source: TalentLMS survey

Providing the ability to easily collaborate, gain skills, and knowledge on the job can also help you scale your team more easily. With the right workflow in place, developers will be able to onboard and get up to speed on new projects more quickly. They can find the information they need without going through gatekeepers, and can share their own insights in a transparent format. A commitment to organization-wide knowledge-sharing ensures that your organization will be able to retain institutional knowledge, even as team members leave over time.

How to improve your DX

So what tools, technologies, and support do your developers need to help them create a fulfilling work culture?

Tools and technologies

With the help of your team, audit the existing tools in the value stream that they use from ideation to deployment. Are there overlapping tools that conflict with one another, cause duplication of work, or create confusion about the correct process to use? Are there gaps in the tool ecosystem where a new tool would help them collaborate better, automate tedious tasks, or smooth handoffs to other teams?

Flexible schedules

According to Stack Overflow's "What makes developers happy at work" pulse survey, the most important factors to developers for taking or keeping a position are salary expectations, the ability to maintain a flexible schedule, and the opportunity to learn on the job.

Room to experiment and learn

While you cannot refactor your application every time you hire a new developer or the most popular language changes, you can make room for developers to experiment with special projects that allow them to flex their skills while solving a business or customer problem. According to Stack Overflow’s 2022 Developer Survey, developers are most passionate about working with the programming language Rust, and want to work in Python and Typescript if they aren't already doing so. Phoenix and Svelte are the most development frameworks, and React is the most wanted. Take the pulse of your own development team to get a sense of which technologies, frameworks, and languages they’re most interested in learning. These learning opportunities can provide opportunities to modernize and streamline how your team works today.

Establish a DX baseline

Once you've analyzed your own development team's priorities, tools, and culture, establish the baseline that you will compare against to validate improvements. Quantitative metrics such as deployment frequency, time-to-production, ticket creation for support requests, and progress in learning new programming languages can help you measure productivity and upskilling or professional growth. Qualitative metrics such as improved retention, reduced absenteeism, more referrals from current employees, improved Glassdoor reviews, and internal NPS surveys can help you measure developer happiness.

Continuous improvement

To see improvement, it's important to optimize your tech stack with the right tools and culture that optimizes efficiency and collaboration. Using a knowledge sharing platform, you can empower your developers to optimize performance through self-service knowledge sharing. Once you've implemented the processes and technologies that you think will support a better DX, track changes in those metrics to see if they're improving over time. Are developers able to produce higher quality code and deploy faster? Are they able to get answers to questions faster with fewer meeting requests? And most importantly, are you improving your metrics around developer retention?

A great DX will help your company scale

By paying close attention to the needs, desires, and goals of your development team, your organization will see advantages in recruiting, onboarding, upskilling, and retaining key talent.

While a competitive compensation package and good benefits are important, developers are equally invested in participating in a workplace culture where they have access to all of the information they need to eliminate roadblocks quickly, and to learn from their peers so that they can continually improve their skills and learn new programming languages. By incorporating knowledge sharing and collaboration directly into your development workflow, you'll be able to empower your development team with the access they need to solve problems and learn new skills from one another.

You'll also be able to preserve valuable institutional knowledge, ensuring that your developers are able to build on one another's work and foster a culture of continuous learning and innovation. When you do see attrition, this ensures that vital information doesn't leave with the subject-matter expert—they'll be able to preserve important context within your knowledge-sharing platform, enabling the rest of your development team to retain knowledge that they can continually build on.

Building a development culture that supports continual, peer-based learning and development will help your development team thrive in their roles.

Your technologists will be able to collaborate across teams, developing a feedback loop that draws in insights from other departments to ensure success in product development and technical support. By building a development culture that breaks down silos across the company, your developers will be able to tap into institutional knowledge instantly, helping them to build and problem-solve faster. Creating a positive developer experience will make your organization a place where talented engineers are eager to collaborate and grow their skills—enabling you to attract and keep the talent that helps your organization thrive.

Integrating Stack Overflow for Teams for a strong DX

In order to develop a best-in-class developer experience, a knowledge sharing platform like Stack Overflow for Teams allows community developers to document context for code, discuss issues, and share knowledge across different teams.

Using a knowledge-sharing platform where they can easily search for relevant keywords and add their own feedback in a vetted and transparent platform gives developers the ability to improve productivity, collaborate more easily, and continually upskill with support from their peers across the organization.

Here is how Stack Overflow for Teams can improve DX for both new hires and senior staff:

- **Speed up onboarding.** When new employees join your organization, they can get instant access to a huge archive of information, including onboarding-specific collections that help them get the lay of the land. From there, they can search for specific keywords to instantly gain context on existing projects so that they can understand the history of what they're working on.

- **Improve productivity and developer velocity.** The platform enables developers to find answers quickly by searching for tags or keywords in an asynchronous format, helping them get unstuck and maintain high productivity without adding distractions to their day. If they can't find the answer to a question, they can ask for support and tag individuals or teams to quickly gain access to knowledge and ensure that it's shared throughout the organization in a transparent manner.

- **Increase employee engagement.** Employees across the organization will be able to share their own unique insights with other teams, and the platform's recognition-based community drives higher engagement and collaboration rates. Because the tool is built on the same platform as the publicly-facing Stack Overflow, which most developers are already using, it provides an intuitive experience that leads to high adoption rates.

- **Eliminate barriers to information.** Stack Overflow for Teams' transparent, centralized knowledge base eliminates barriers to finding and contributing knowledge throughout the organization. Its transparent Q&A format promotes sharing of institutional knowledge that might be lost when a key SME leaves, and its community voting system makes it easy to find the most relevant and valuable information on a given topic quickly, based on the votes of fellow team members.

stack overflow
FOR TEAMS

Ready to improve your developer experience with better collaboration that breaks down silos and removes roadblocks to solving complex technical challenges?
Learn more about Stack Overflow for Teams.
(It's free forever for groups of up to 50 users.)