

Bringing Stack Overflow for Teams into your organization

In this guide, we've compiled best practices from organizations of all sizes that have successfully brought in *Stack Overflow for Teams*.

Their insights and experiences will help you with the process of building internal support and securing funding for *Stack Overflow for Teams*. This blended approach includes inputs from large multinational corporations with more than 50,000 employees alongside insights from rapidly expanding organizations with fewer employees but high growth goals.

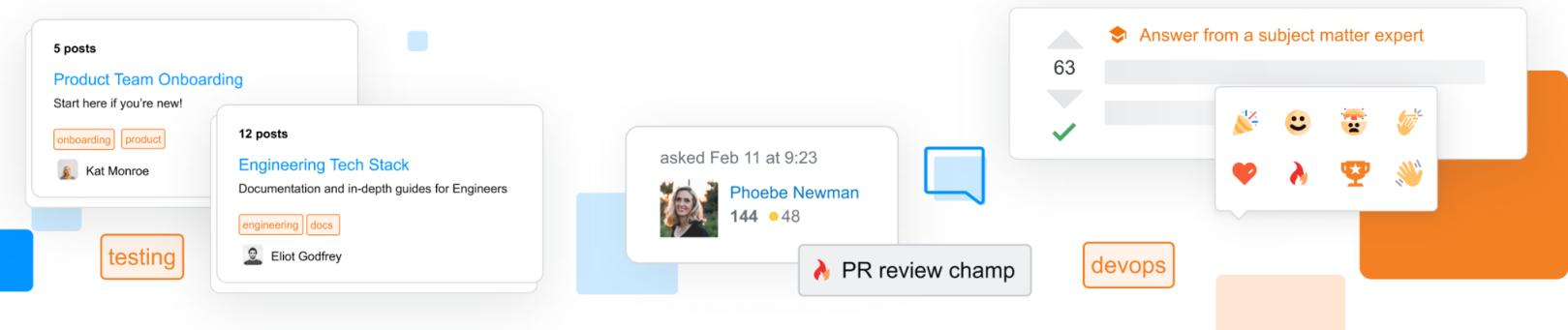


Table of contents

Step 1: Know the problem you need to solve

Step 2: Collect data

Step 3: Find your core group

Step 4: Start getting broad buy-in

Step 5: Enlist an executive sponsor

Step 6: Understand your purchase process

Step 7: Build your requirements list

Step 8: Consolidate input and requirements

Step 9: Gather and evaluate solution partners

Step 10: Plan your business case/request

Invest effort now to reap the rewards later

Step 1: Know the problem you need to solve

Bringing in any new platform or tool takes some pre-work. Following these steps to build the case for Stack Overflow for Teams makes it easier to show the value it will bring to your team and organization.

The first step starts with having a clear idea of the business challenge you need to solve.

Common reasons organizations bring in Stack Overflow for Teams

Why do organizations bring in Stack Overflow for Teams? It's not only about knowledge management. Right now, companies face huge, often nebulous challenges related to onboarding and retaining the right people, improving productivity without sacrificing the developer experience, and supporting innovation.

Here are the business challenges that lead organizations to consider Stack Overflow for Teams:

Onboarding and retaining talent

Ineffective onboarding processes have negative ripple effects that can spread throughout your organization. These include slower time to ramp, an increased burden on SMEs, and diminished productivity from the people who are supporting the new hires.

Stack Overflow for Teams helps with this by:

- ✓ Improving the time to ramp for new employees and helping to upskill existing employees
- ✓ Shortening the ramp time to productivity and improving developer satisfaction

Improving learning and growth opportunities

Developers, like all employees, want to work where they are valued and where they have opportunities to learn and grow. [More than half of developers we surveyed](#) said access to growth opportunities was a priority in deciding whether to stay in their current role or find a new job. Too often, knowledge and SMEs are siloed.

It's hard to identify and recognize SMEs, especially if they work outside your immediate team. Devs don't have visibility into what's happening beyond their team; that makes effective cross-functional collaboration difficult. In the absence of a knowledge-sharing and collaborative culture, it's tough for developers of any level or tenure to upskill/reskill.

Stack Overflow for Teams helps with this by:

- ✓ Upskilling talent by connecting knowledge seekers with knowledge experts
- ✓ Identifying new SMEs, uncovering skill gaps, and allowing organizations to allocate top talent across different roles
- ✓ Recognizing top contributors elevates employee engagement

Improving productivity and preventing knowledge loss

Autonomy is important to feeling productive. It's frustrating for developers to have to stop what they're doing and burn time hunting for SMEs or solutions to problems they could solve themselves if they had access to the right information. Constant distractions, interruptions, context-switching, repeated work, and knowledge loss takes people out of their flow state and creates a poor developer experience.

Stack Overflow for Teams helps with this by:

- ✓ Enabling developers to get unstuck and back to work quickly
- ✓ Providing an intuitive experience and integrations that improve adoption and reduce distractions
- ✓ Eliminating barriers to finding and contributing knowledge
- ✓ Fostering community crowd-sourcing to drive greater collaboration across distributed teams
- ✓ Quickly identifying the most trusted content and most relevant expertise

Accelerating innovation

Dismantling silos and connecting people with the information they need when they need it accelerates innovation, keeps tech debt to a minimum, and speeds up the adoption of new technologies. The organization wins when it frees developers to spend more time and energy on higher-order work.

Stack Overflow for Teams helps with this by:

- ✓ Accelerating engineering innovation through access to the right knowledge in a timely fashion
- ✓ Providing instant access to expert, relevant knowledge that fuels innovation and collaboration
- ✓ Enabling crowdsourced community participation resulting in more creative solutions
- ✓ Delivering data-driven insights to identify barriers and obstacles that prevent innovation
- ✓ Offering a familiar UX that reduces time engineers spend tracking down answers across various channels

Naturally, most customers don't come to us looking to address a single business challenge or use case. These business challenges aren't mutually exclusive; they frequently exist in tandem. The initial reason for adopting Stack Overflow for Teams might be to improve or enhance productivity, but organizations will quickly see the potential for improvement in other areas, too.

The core benefits of the platform have a long reach. A small improvement in one area can create positive ripples that inspire improvements in other areas of the business. It starts with focusing on the developer experience (DX), from a developer's first day on the job through their entire tenure at your company. Creating a strong developer experience means making sure devs feel productive (which is different from measuring their productivity), ensuring that they feel connected to peers and SMEs throughout the organization, and offering new learning and professional growth opportunities.

Getting the developer experience right has a positive impact on business-critical metrics like productivity, time to market, and time to resolution that, in turn, fuel higher-level improvements: more customers, more sales, a better position in the market, greater overall profitability—all because you're taking proactive steps to reduce costs and inefficiencies, including developer turnover.

Step 2: Collect data

Making the case for Stack Overflow for Teams requires that you quantify your current state to determine your future state. You need to understand where you are today in order to determine where you want to be tomorrow. What metrics do you have that you can use to illustrate the hardship of your current state? Here are some of the metrics our customers have used to quantify their problems:

- Number of support or team channels used across programs like Slack and Microsoft Teams
- Number of messages posted in Slack/Microsoft Teams channels
- Number of places where knowledge lives (wikis, docs, repositories)
- Emails sent to a distribution list or ticketed
- Length of threads/comments happening within project management system
- Average time to onboard a new hire
- Number of people supporting onboarding new hires
- Number of new hires expected
- Number of tenured employees lost
- Number of orphaned projects, programs, and/or documentation
- Productivity metrics, such as [DORA metrics](#)

Find out where you can pull these metrics - you'll likely need to connect with colleagues in different organizations for some of them (e.g., number of new hires expected).

If you don't have metrics, you can build them yourself. A short survey is usually sufficient to establish a baseline.

We recommend using the questions below. They match what we asked in our 2022 Developer Survey. You can ask your teams the same questions and compare their answers against the larger developer audience.

- ✓ We have packaged a survey and some analysis tools for you to use. If you are interested, [contact us](#) and we'll send it over.

What team/organization are you in?

Enter name of team/org (or you can provide a list)

Are you an independent contributor or people manager?

- ☐ Independent contributor
☐ People manager

How many years of working experience do you have?

Enter number of years (round up to the nearest year)

Please rate your level of agreement with the following statements

Strongly disagree, Disagree, Neither agree nor disagree, Agree, Strongly agree

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
I have interactions with people outside of my immediate team.					
Knowledge silos prevent me from getting ideas across the organization (i.e., one individual or team has information that isn't shared with others)					
I can find up-to-date information within my					

organization to help me do my job.					
I am able to quickly find answers to my questions with existing tools and resources.					
I know which system or resource to use to find information and answers to questions I have.					
I often find myself answering questions that I've already answered before.					
Waiting on answers to questions often causes interruptions and disrupts my workflow.					

How frequently do you experience each of the following?

Never, 1-2 times a week, 3-5 times a week, 6-10 times a week, 10+ times a week

	Never	1-2 times a week	3-5 times a week disagree	6-10 times a week	10+ times a week
Needing help from people outside of your immediate team.					
Interacting with people outside of your immediate team					

Encountering knowledge silos (where one individual or team has information that's not shared or distributed with other individuals or teams) at work					
--	--	--	--	--	--

On an average day, how much time do you typically spend searching for answers or solutions to problems you encounter at work?

This includes time spent searching on your own, asking a colleague, and waiting for a response.

- ☐ Less than 15 minutes a day
- ☐ 15-30 minutes a day
- ☐ 30-60 minutes a day
- ☐ 60-120 minutes a day
- ☐ Over 120 minutes a day

On an average day, how much time do you typically spend answering questions you get asked at work?

- ☐ Less than 15 minutes a day
- ☐ 15-30 minutes a day
- ☐ 30-60 minutes a day
- ☐ 60-120 minutes a day
- ☐ Over 120 minutes a day

How frequently would you say you visit Stack Overflow?

- ☐ Never
- ☐ Less than once per month or monthly
- ☐ A few times per month or weekly
- ☐ A few times per week
- ☐ Daily or almost daily
- ☐ Multiple times per day

Step 3: Find your core group

Enlisting allies to help make the case for adoption increases your likelihood of success. Talk to your peers about their work headaches and find a core group of people who are experiencing the same challenges that you are (refer to Step 1). Ask them how, in an ideal world, they would fix these challenges. Don't worry about collecting solution requirements at this point; just find out what they need to alleviate their pain points. Ask if they'd be open to working with you to find a solution to these problems.

Step 4: Start getting broad buy-in

As you assemble your core group of advocates, start identifying other people in the organization who might be encountering similar roadblocks. Reach out to those folks and explain that you've identified a problem and you're collecting input on how to fix it in order to evaluate potential solutions. If they're willing, meet with them and get their thoughts. Below is a rough discussion guide you can use for that conversation:

- Do they agree that the challenge you've identified is a problem? What other challenges come to mind?
- Tell them what you've identified, based on what you've learned from your core group, as ideal fixes for the problems you face. This part of the conversation is important: it gives people who don't have as much context a framework for considering ideal solutions.
- In an ideal world, what else would they have to address these challenges? What does the world look like after they've addressed these challenges?
- Would they be open to working with you on a solution to these problems?

Keep in mind the importance of getting broad buy-in: that is, buy-in from multiple teams across the organization. In most cases of software adoption, starting small is the best practice, but the opposite is true of Stack Overflow for Teams. Since it's community-driven, the platform is more effective and valuable when multiple teams are involved from the beginning. Restricting the initial rollout to a small group won't allow you to show the full value of Stack Overflow for Teams. For example, cloud teams and SDLC/platform engineering teams often serve as support for other technical teams. If there's no buy-in from support teams or they don't have access to Stack Overflow for Teams, the benefits are minimal: both from a knowledge-sharing and a knowledge-seeking perspective.

Step 5: Enlist an executive sponsor

We recommend identifying someone in your organization's leadership who can serve as an executive sponsor and high-level advocate for your adoption. This person can help ensure that your proposed implementation aligns with your company's strategic goals, assist in building support and overcoming resistance from other executives, and otherwise provide support and direction as the adoption goes forward.

An executive sponsor should understand and agree with:

- What problem(s)/business challenge(s) you need to solve
- How Stack Overflow for Teams can solve those challenges, and how that positive change will impact the organization
- Who's involved and how they'll be supporting the adoption
- How implementation will affect employees, consultants, and customers

Step 6: Understand your purchase process

Another way to increase your chances of success is by understanding your organization's purchase process. What's involved in that process? Who do you need to loop in and when? Often, legal and/or procurement requirements are afterthoughts, but bringing these teams in early to collect their requirements saves time and friction—and helps build that broad buy-in you need.

Step 7: Build your requirements list

Now it's time to collect and agree on the requirements the solution must have. Solicit feedback from your advocates and ask your core group to suggest other requirements. You can categorize these simply as "must-have" or "nice to have," or apply a more granular, weighted system to your list. The below requirements list is a good starting point:

Functional requirements

What types of content are available (e.g., Q&A, long form)?

How is the content organized?

How is knowledge captured and added to the system?

How much effort is required to add content to the system?

Note: This is an especially important capability to evaluate. Consider how many minutes it will take people to add content to the system.

What kind of individual/team should moderate (for example, by deciding how to manage tags)?

- ☐ Does the platform include technology to reduce or eliminate duplication/clutter?
- ☐ Does the platform have robust search and discovery capabilities?
- ☐ Does the platform have a single owner or multiple owners?
- ☐ Does the platform offer support for content curation (tagging, taxonomy)?
- ☐ Does the platform have mechanism(s) to support content freshness (e.g., answer

owner workflow to refresh or take down after a given period of time)?

- ☐ Can the platform support separate project-specific interactions (for example, employees using a more technical product like an API or SDK)?
- ☐ Does the platform include adoption catalysts (gamification, community-building aspects)?
- ☐ Does the platform describe effective rollout/adoption strategies/patterns?
- ☐ Does the platform provide insight into roadmap and product health: Number of releases this quarter/year, number of planned releases, quality of releases?
- ☐ Can new content be pre-populated to jumpstart the creation of more material and drive adoption?
- ☐ Does the platform describe the people and specific skills necessary to effectively moderate the solution?
- ☐ Is the platform globally accessible for all teams that need to use it, both as searchers and as contributors?
- ☐ Is the platform transparent to the entire organization?

Integrations and what you need the integrations to do as part of the workflow

- ☐ Microsoft Teams
- ☐ Atlassian Jira
- ☐ Atlassian Jira Service Manager
- ☐ Atlassian Confluence
- ☐ Salesforce
- ☐ IDEs (VS Code, IntelliJ)
- ☐ GitHub
- ☐ API/SDK

Analytics

How are they measuring velocity of use (how quickly the system is being adopted and how fast it's growing)

Is the platform enabling knowledge reuse?

How are they measuring knowledge creation & curation (e.g., content organization, timely responses to requests for help, content is accurate and trusted, low number of unanswered requests for help, participation is consistent)?

How are they measuring engagement metrics (e.g., unique contributions, the velocity of contributions, time to the first contribution, searches, etc.)?

How are they measuring knowledge diffusion ratio (e.g. cross team knowledge flow, link sharing, content referrals)?

What other metrics do they provide that can ladder up to our business metrics?

Security/compliance

- ☐ Single sign-on (SSO) integration
- ☐ Active Directory Federation
- ☐ Supports different role types (which ones do you need?)
- ☐ Current compliance/security audits (such as SOC 2)

Service delivery

- ☐ On-premises option(s)
- ☐ SaaS option(s)

Services

Describe professional services offering/packages required for successful implementation/rollout

Describe typical cost of implementations similar in size to the one under consideration (250-500 initial users; scale up to 3,000+)

Solution partners

The current market share and user base (domestic and international)

Core users' familiarity of the platform

Expertise in specific problem area

Wealth of resources available to educate yourself and others

Cost

How many users will use the solution? Can the solution scale to the rest of the company?

Pricing model: flat, tiered, discount(s) available

Example pricing for 500 users (license, recommended professional services, SSO integration)

Infrastructure, if on-prem

Step 8: Consolidate input and requirements

Once you've collected your requirements, formalize them in a document or slide deck and share it with the people you met with to make sure everyone's on the same page. Identify the individuals you spoke with and the size of their teams to communicate the scale of the problem (and the solution).

Step 9: Gather and evaluate solution partners

The next step is to assemble an initial list of solution partners. Do your research and ask for feedback and input from the teams you've been talking to. Once you have your shortlist, schedule evaluations to help you identify the best fit. We recommend planning on at least three meetings before picking the top two partners. Keep these considerations in mind as you go into those evaluations:

- Is the solution purpose-built to address the factors that are limiting your ability to meet business goals?
- Does the solution meet your list of requirements? Does it just check the box of a requirement or does it address the problem(s) you are trying to solve?

✓ Is legal or procurement part of your organization's purchase process?
If so, involve them in the evaluation process.

Your evaluations can follow this framework:

1. Initial call
2. Demo
3. Metrics discussion: What metrics do they offer? What metrics do you need to demonstrate value?
4. Follow-up call to align on decision criteria, proof points required, and action items.

Consider recording these calls for people who can't make the meeting. Sharing information asynchronously is a great way to keep folks informed without causing distraction.

Step 10: Plan your business case/request

Leadership wants to know why, from a business perspective, they need to fix this problem and how you'll make that process a success. Keep communication clear and impactful by designating one person to lead the business case presentation while the rest of the team chimes in where it makes sense. The business case needs to present Stack Overflow for Teams as solving a business challenge. On a high level, Stack Overflow for Teams needs to check at least one of three boxes for your organization:

- ☐ Increases revenue
- ☐ Decreases cost
- ☐ Reduces risk

Each of these boxes speaks to the bottom line leadership needs you to address: How does implementing Stack Overflow for Teams benefit the business?

It's also helpful to align your request with high-priority initiatives and broader organizational goals that are near and dear to the leadership's heart. For a bigger company, these goals might include tech stack modernization and cloud migration; for smaller, earlier-stage companies, the goal might be onboarding new people quickly and efficiently to allow the company to scale to 2x.

This is where your executive sponsor can really help you. They can provide guidance on best practices and requirements for your organization. They can tell you which stakeholders need to be in the room. They may be able to soft pitch the idea to their peers and get feedback that can help you create a stronger business case.

Let's zoom in and consider the structure your request should follow to maximize your chances of success.

- ✓ We have packaged a deck template for you to use. If you are interested, [contact us](#) to set up a time to speak with someone.

Identify the business goals

Start with a quote from an executive to tie your request to leadership's priorities, identify the problem you need to solve, then start outlining the business goals Stack Overflow for Teams can help you meet.

Identify the factors that contribute to the problem you need to solve - use the data you collected to prove your point

- ✓ Frequent interruptions
- ✓ Non-scaled SMEs
- ✓ Hidden or inaccessible SMEs
- ✓ Massive hiring goals
- ✓ Employee turnover
- ✓ Scattered knowledge
- ✓ Lost knowledge
- ✓ Out-of-date knowledge
- ✓ Lack of clear ownership over knowledge
- ✓ Redundant questions/needs
- ✓ Friction when moving fast
- ✓ Reskilling/upskilling challenges
- ✓ Need to innovate and remain competitive

Establish that challenges are significant but solvable

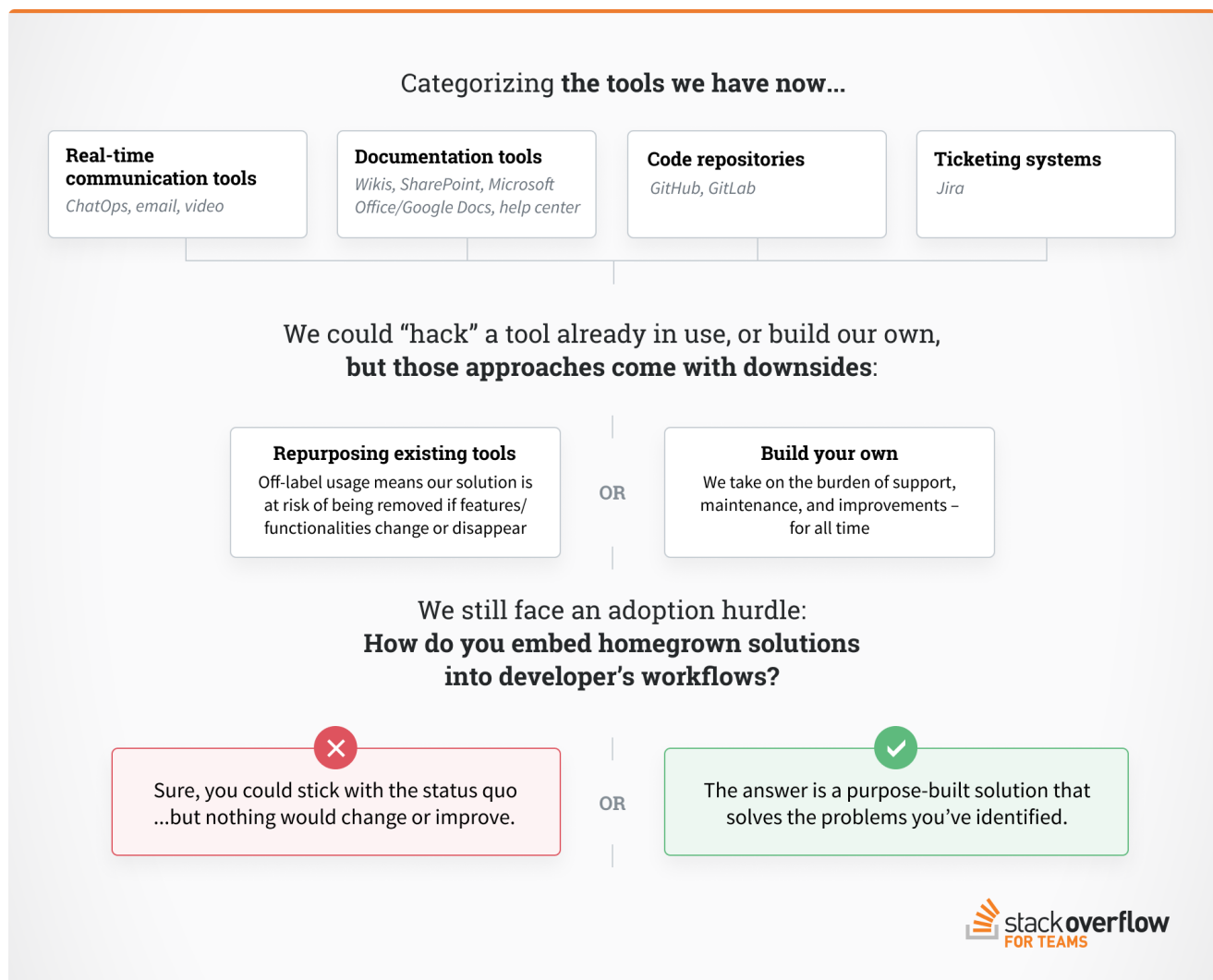
Show leadership that these problems are significant enough to keep you from meeting business goals but still solvable, in the sense that there are specific things you can do with Stack Overflow for Teams to solve the problem. Address the immediate issues and their ripple effects on the business (see the table below). The more quotes and specific data points, the better: they make the pain points real and quantify the impact of these blockers on the business.

What's happening now	Why it's hurting us
Tech debt limits the pace of assimilating innovative technologies	Slow time to market for new products, features, and services
Knowledge silos hinder the collaboration and reuse of knowledge required to deliver projects	Multiple teams are spending time doing what has already been done
Identifying experts and connecting them to solutions is difficult	Inability to unlock expertise leads to lower-quality code and products
Redundant questions and frequent interruptions hinder engineers from building products	SMEs are spending time on redundant questions instead of focusing on high-value project delivery
Current collaboration process (ChatOps, wikis) is broken and leads to lengthy development cycle	Loss of first-mover advantage due to lengthy development cycles
Developers don't know where to go or which SME to ask to find the information they need	Lengthy time to search for and find information
SME are constantly interrupted with repetitive questions, preventing them from working on strategic projects	SMEs spend time on repetitive questions and support requests instead of releasing new features
Daily repeat questions that don't easily surface and retain knowledge	Time wasted building one-off or redundant solutions
Lack of centralized knowledge: Finding and sharing information across multiple data sources is complex and inefficient	High number of sprint cycles ⇒ slower time to market (deployment velocity)
Difficult for developers to adopt new technologies because documentation is missing or not consumable	Missed timelines to deliver software or to adopt new technologies
Losing knowledge as talent walks out the door	Wealth of knowledge retained/lost; lengthy time to pick up where lost employees left off
Not enough talent to fill the roles: everyone is competing for the same talent	High friction to assimilate new technologies

Limited performance management for top talent (SME for emerging areas)	Losing developers to the competition
Ineffective onboarding processes (how to get answers from SMEs, where to learn what you need to learn)	Lengthy time to onboard new employees and extended burden on colleagues supporting new hires
Poor developer experience: engineers are not focused on areas that they enjoy and allow them to grow	Low developer productivity
Innovation is hindered	Adoption rate for new technologies is slow; developers have less time for creative, higher-order work

Show that current measures/tools aren't solving the problems

The next step is crucial: Show leadership that current measures/tools aren't solving these problems. Copy this illustration and use it your presentation.



You have a plan

You've compiled a list of functionalities to help you pick the right solution. Share your requirements list and explain each of the categories you evaluated, from functional requirements to integrations, service delivery, and cost. Explain, specifically and with plenty of examples, quotes, and data points, how Stack Overflow for Teams solves the problems your organization faces and how you'll measure the platform's performance after adoption.

You've brought in the right people to help you implement the solution

Prove to leadership that you've identified the people who will help you make this implementation a success:

- ✓ The broad group of people you spoke with
- ✓ Your executive sponsor
- ✓ SMEs (individuals and teams)
- ✓ Champions/advocates/ambassadors on each team
- ✓ Data analytics/business intelligence (optional)

You have metrics for monitoring success ready to go

It's important to show leadership that you already know how you'll monitor and evaluate the implementation you're proposing. Data is your friend: Pinpoint system metrics that will measure success and ROI metrics to highlight business value. Also, bring out the data you uncovered to quantify the pain points your organization is dealing with. Demonstrate to leadership that you're making this new solution a measurable part of your team's goals.

You have a plan for adoption and ongoing usage

Increase leadership's confidence in your proposed solution by sharing a plan for:

- ✓ Adoption support
- ✓ A comms plan for rollout
- ✓ Managing integrations and behavior shifts

Having a plan for supporting adoption, rollout, and ongoing usage helps mitigate the risks of shelfware and low adoption. Your solution provider should provide you with a robust plan that you can include here.

Ask for what you need to make it happen

Communicate what you need to make implementation a success:

- ✓ Executive support
- ✓ Time allocated to building/launching core team
- ✓ Budget: POV and then full deployment

Ask for what they need to know

- ✓ Answer questions and flag any unexplored topics for more in-depth review
- ✓ Recommend that leadership meets with the solution partner

Invest effort now to reap the rewards later

Showing your executives and internal partners that Stack Overflow for Teams is a sound investment takes some upfront work, but the rewards are entirely worth it. Start by gathering data on your specific pain points, cultivating broad buy-in across teams, recruiting an executive sponsor, and deciding on the metrics you'll use to see how well the solution is performing.

We have helped thousands of organizations like yours see success with Stack Overflow for Teams. Read through our [case studies](#) or [reviews from G2](#).

When you are ready, [contact us](#) and we'll be your partner to help you navigate the process.

