



---

# How to build an effective developer relations team

# The DevRel Collective

By Quinton Wall

# CONTENTS

Developer Relations is a fascinating field. Typically straddling Product and Marketing, it needs to balance the goal of growing the developer adoption funnel with the requirement to teach developers how to use your product or platform. Where things differ from traditional marketing, in particular, is the way in which you do it.

- > **How is Developer Relations different from Product Marketing?**
- > **Typical Developer Relations Tasks**
- > **Blogs and Content**
- > **Code Samples & Snippets**
- > **Case Studies / Blueprints**
- > **Tooling**
- > **Learning Paths**
- > **Events**
- > **Branding**
- > **Community**
- > **Social Media**
- > **Growth Engineering**
- > **Metrics**
- > **Conclusion**

## How is Developer Relations different from Product Marketing?

> Traditional product marketing leans heavily into strategies such as demand generation, retargeting, SEO, and awareness. Once someone signs up for your product, they may continue to reach out via nurture flows, newsletters, and events or webinars to keep users up to date on the latest news and features., but that is about where it ends. Developer Relations primary job, on the other hand, is to take a technical audience and educate them on how to integrate your product into their solution, or in the case of a platform company, show developers how to built atop the platform using best practices.

In short, Product Marketing is more about awareness and adoption. Developer Relations is about enablement and retention. The lines get a little more blurry in larger organizations where Developer Relations likely has personal to perform their own marketing and demandgen functions.

**Next, we will cover the typical developer relations tasks**

## Typical Developer Relations Tasks

### Blogs & Content

> Technical blogs describe a particular feature, new release information, sharing best practices, or how a product integrates with a partner technology or latest hot technology device or framework. They are the bread and butter of Developer Relations content. Blogs should contain technical how-tos, use different programming languages based on target audiences, and be tailored to specific geographical regions to increase relevance.

### Code Samples & Snippets

> Every piece of Developer Relations content should contain code samples and snippets. Developers want to test your product, solve a problem, and increase their productivity using your service. Use tools like GitHub and CodeSandbox to make it incredibly easy for them to take a sample and run it. I'm also a big advocate for a canonical sample application that demonstrates a comprehensive solution using your product or platform. This sample can be an invaluable guide for developers to not only get started with your product, but learn the more advanced topics and features.

## Typical Developer Relations Tasks

### Case Studies/Blueprints

> Case Studies are an excellent way of harvesting the best practices of customers. Developers are betting on your product and appreciate hearing from other developers who have used it. Take care with your case studies to ensure they have sufficient technical substance though. Developers want blueprints and architecture diagrams vs. statistics of time saved, cost to maintain etc. Leave the statistics to product marketing case studies. Give Developers the HOW, not the WHY.

### Tooling

> Perhaps the most important thing Developer Relations can do for their community is to create tools for Developers to improve the experience of using your product. This could be API wrappers, SDKs, playgrounds and sandboxes, CI script, you name it. The best functioning Developer Relations team are often slightly out front of the Product teams to ensure Developers have what they need to get the job done. Developer Relations can often work more rapidly than Product teams because a toolkit may not require official SLAs or support agreements that Product must factor in.

A great Developer Relations team can provide invaluable feedback to Product on the direction of future releases and features, often rolling in toolkits and wrappers into later, officially supported product features. One great example of this is the Salesforce IdeaExchange. During my time at Salesforce, the Developer Relations team often worked with community members to create wrappers based on ideas in the Idea Exchange, and then championed official support internally with Product Management.

## Typical Developer Relations Tasks

### Learning Paths

> Learning Paths provide getting started guides and tutorials designed to guide Developers in their journey to learn your product or service. They typically start with the basics and increase in scope and complexity to cover all of your technical offering. Developer Relations will spend significant time creating tutorials designed for onboarding, ecosystem enablement, and solution design all based on the goals of the company and developer adoption. These activities should go hand in hand with user journeys created either by Product Management or Developer Relations. Along with blogs, learning paths are likely the most common content Developer Relations will create. Depending on the size of your organization, Developer Marketing, or Growth teams will leverage this content into onboarding nurture flows, and even in popular MOOCs such as Coursera or Codecademy.

### Events

> Events play a large part in the Developer Relations team. Advocates will be presenting at company or industry events, conducting workshops and webinars, and often collaborating with Sales teams on enablement activities for strategic customers. Events are obviously an important aspect of awareness, but care should be taken that content create for events must be designed to be reused across different channels including learning paths, analyst activities, blogs, code samples etc. Too often, Developer Relations can spend significant time performing one-off activities due to events. Avoid one-off content wherever possible.

## Typical Developer Relations Tasks

### Branding

> I'm a huge advocate for Developer Relations to own the Developer Brand. This must be done in collaboration with Corporate and Product Marketing obviously, but Developers are attracted to products and communities that have a strong vibe about them. Heroku and Twilio are two excellent examples. Developer branding can be used on stickers, t-shirts, websites, events etc - anything that developers touch and experience that makes your product stand out. Don't underestimate the importance of branding to developers. Invest time in darker color palettes, a cool mascot, tone of voice/writing, and inject it everywhere. Developers should know your product simply by its icon.

### Community

> Community is everything for growing Developer adoption. The obvious aspect is as a form of support for developers to ask questions via discussion forums, but community can, and should, encompass things such as open source initiatives, events, job resources, and more. Technology changes so rapidly that simply basing your Developer engagement and retention of say a specific language such as Ruby, or a design pattern such as SOA, is a risky proposition. Building a community around Design, AI, or Mobile Development on the other hand is far more encompassing and allows you to constantly inject new languages, frameworks, trends and much more.

## Typical Developer Relations Tasks

### Social Media

> Depending on the maturity of your company, typical social media such as Facebook and Twitter/X may be handled by another department other than Developer Relations, or you may have a dedicated developer focused social media handle. Social media is great for broad awareness and prompt response to community questions. Developers are also heavy users of Twitch and YouTube for educational content. Developer Relations teams should actively build content for these channels and partner with Demandgen teams to slice up this content for retargeting ads and short form content suitable for TikTok.

### Growth Engineering

> I don't see this owned by Developer Relations teams too often, but growth engineering activities can include modifications to the signup or onboarding process to add a guided wizard to customize developer landing pages in products, Pendo flows for particular personas, and tweaks to nurture flows. Typically such activities are either owned by Product Management or dedicated Growth Engineering teams, with heavy input from Developer Relations.

If your Developer Relations teams is mature enough though, I encourage adding one or two growth engineers to the team dedicated to build in-product, or adjacent-to product based solutions designed to increase developer conversions. An excellent example of this model was the creation of the Twilio CodeExchange. The goal of the Growth Team when launching this feature was to make it easier for developers to get started and quickly realize the value of Twilio, thereby incentivizing them to upgrade from the free trial.

## Typical Developer Relations Tasks

### Metrics

> The final item in the list is ensuring that Developer Relations team have access to the right metrics to constantly measure the health of the Developer community. Are signups growing? How long does it take for a developer to make their first API call, or go into production? What's that MAU of developer? Is it increasing or decreasing? How many hours of video was watched via YouTube? Do you have a retention or activation problem? What are developers in your community doing or saying, and how can you engage with them? What's the number of support tickets or forum questions raised vs answered? And many more.

This idea of defining the right metrics and using them to identify friction points in the developer journey are perhaps the largest part of any consulting engagement I do for clients. Strong Developer Relations teams use these metrics to focus their activities on what is most important at this point in time.

# Conclusion

---

Developer Relations is a fascinating mix of product, marketing, coding, growth, and smattering of acting all designed to improve the developer experience for your users. The activities above are the most common tasks I have seen, or been apart of, for Developer Relations teams to focus on. Depending on the size of your team and organization, some of these may fall outside of the Developer Relations team, but require close collaboration. The key to working with Developers is to show them how to use your product. If you can do this successfully, and your product solves their problem in an efficient and modern way, you are a giant step forward in being successful build a vibrant developer community.

If you'd like help getting your Developer Relations program up and running, reach out to us at [thudevrelcollective.com/get-started](https://thudevrelcollective.com/get-started)