

Top 10 Issues For Developer Relations Managers

By [Mike Slinn](#), CTO & Founder of [Zamples, Inc.](#)



There is no "School of Developer Relations" that one can attend to learn about how to manage a developer relations program. Having participated as a developer in many developer relations programs, and having been involved in setting up others, I have seen a few shining examples of how to do it right, and many more marginal developer relations programs. So that you don't have to stumble through the process while trying to figure out the issues for yourself and your company, I've put together this brief introductory article for you.

Software platform vendors sell software that must be customized and integrated. Because of this they have an extra mission beyond simply selling software licenses - growing and servicing a developer community. An effective developer community multiplies the reach of the software platform vendor by leveraging the total community resources towards the software vendor's goals. In short, the vendor gets a big result from a small investment.

Developer relations programs create and nurture the developer community. Not only do channel partners bring the vendor's products into specialized markets, thereby increasing sales, but also members of the developer community interact with each other, which improves technical support quality and decreases costs.

Properly championed and managed, a developer relations program is an extreme example of how to achieve dramatic results by leveraging limited resources. It is remarkable what a two-person developer relations team can accomplish!

Here are ten important issues for developer relations managers of software platform vendors:

1. **Enlist senior management support.** Developer Relations groups are usually placed under the purview of the marketing department, but it's not a perfect match. Developer relations personnel interact closely with the engineering, technical support departments, and must be able to service hard-core developers. The key issue is to continuously evidence to senior management that the developer relations team is a strategic initiative that generates both long-term and short-term benefits, and isn't just a cost center handing out freebies.
2. **Know your developers.** Your external customers are developers, and you must know what their technical problems and issues are. Developer relations is also a conduit to sales, for they open up accounts that would otherwise not present themselves. Treat your developers as customers, by tracking their activity, providing mechanisms for feedback, local contact points, and regular briefings.
3. **Grow the community.** This is where your organization enjoys the benefits of the massive fan-out that an active developer community can provide. For example, 1000 developer companies might well have 5000 developers and 7500 sales people... all managed by your tiny developer relations department! Properly managed, a developer relations program can be an effective means of driving sales, as well as reducing the cost of technical support.
4. **Offer a tiered program.** Not all developers are created equal. Smaller developers will generate smaller sales, but there are more of them. Innovation comes from all sizes of companies. Be

flexible - offer a free developer program, and for-fee tiers for developers who will make more demands.

5. **Profit or cost center?** Developer programs drive sales indirectly and can take a year or two to produce tangible benefits. Developer relations should be considered as a strategic initiative that generates far more revenue than it consumes in resource. You'll need to survey your partners and customers to accurately measure the sales benefits of your developer program. If properly championed and administered, developer relations has the potential to consistently generate the single largest return on investment of any department in your company.
6. **Give developers what they need.** Developers need prerelease and released copies of your software, technical articles and code examples. One of the pioneers of the computer industry, a multi-billion dollar a year company, devotes a significant percentage of their engineering resource to creating code examples. The importance of code examples cannot be over-emphasized - Evans Data's 2002 annual developer survey shows that code examples were the single most important issue for developers.
7. **Set goals.** Developer relations teams must be proactive in order to be effective. You must regularly evidence your progress to your stakeholders: management, the development community and the other departments within your company.
8. **Stay in touch.** Just like the sales staff, developer relations personnel need to stay in touch with their community. Automation helps, but machines can't replace interpersonal relations. It's better and easier if the developers have access to an online community where they can interact with each other and selected staff.
9. **Rotate your staff through developer relations.** There is no better way for your staff to understand your developer community than by working with them a few days every quarter. Enlist people from sales, engineering, technical support, technical publications and senior management to work with you on a formal schedule. Obtain a commitment from each participant for one day a month. Keep a blog on your intranet for you and the participants to publish their activities. It's amazing what cross-fertilization will accomplish.
10. **Extend your enterprise.** Provide a means for your developers to be able to access key people in your organization easily. This is an area where automation (like CRM or workflow) can really help. Asynchronous accessibility is often key, such as provided by email, but simply publishing email addresses is not a viable solution. Developers need to be able to discover information, make product suggestions, obtain workarounds for bugs and collaborate on sales opportunities in an efficient manner.

Developer relations programs are all about the effective leverage of limited resources. You aren't babysitters, worker bees or receptionists, so outsource and automate your activities to the maximum extent possible. Concentrate on growing an effective developer community, and plugging them into the rest of your enterprise. The results will be astounding!

Mike would like to [hear your comments](#).