**5 Mistakes Developers make Selling to Developers**

<https://www.47hats.com/2010/08/04/5-mistakes-developers-make-selling-to-developers/>

A few typical things can happen when developers sell and market development tools to other developers. Here are some things to be careful about:

**Overselling**

OK, here is the deal.  It’s plain and simple, but sometimes forgotten.  Developers, engineers, and any scientific thinking person tend to reject outlandish marketing that overpromises.  Also, that same disgust and rejection is now directed towards “management speak”. How many sites, sales processes, and emails to prospects are still bathed in sports analogies, non sequitur solution selling, and ROI vagueness, It’s simple, in reality developers just want products to make their life easier.  Keep away from sensationalism and overstating the benefits of your product.  Keep it pragmatic, honest, direct, and understated.  Solicit both good and bad feedback.   This goes not only for marketing, but also for communication during the sales process.

Also, although you might think it is about closing, it is a sales person’s job to help discover issues.  As far as development tools go, all tools have issues.  All tools have problems.  Not sure if I should write this, but it is your job is get users to come to conclusion that your tool “sucks less”.  Even if you are lacking some features, if you are responsive and helpful, you can land a happy user.

**Remove roadblocks**

Make it as easy as possible for someone to get a look at your product.  As a general rule, remove roadblocks for trying your product.  If you can get the user experience in a web based sandbox without an install; do it.  If you have several registration pages with logins to get a build; remove it.  Make it as easy as possible for new users to get a look at the product and make it even easier to solicit feedback.  If you can solicit feedback from within the product; code it in, as it is much easier than soliciting an email response (if you capture email information).  You will have very little time to hold a prospective users attention, so make the most of the situation.

**We get it.  You’re smart. We’re smart. Now about the order?**

Monitor situations where the support and development team gets into conflicts over technical details.  Is your team responding like a cheery English mouse or more like the soup nazi from Seinfeld?  Sales can and should help monitor the situation.  Think about it, for every 1 response you get about an issue, there could be 5 to 10 silently suffering.  Make sure you aren’t dismissive of technical objections on the way to an order.  Make sure the response from your company doesn’t make them look technically challenged.  After all, they are possible customers. The company goal is to sell product, not to show how smart we are, although we sometimes can get confused.

**Be responsive.**

Think of your product as being completely like an open source project without the need for contributors to do the hard work of coding features.  Get responses.  If someone makes a feature request; respond.  Listen, ask more questions, survey, and solicit responses.  If you are able to turn those responses into product changes, you’ve empowered those users into loyal users.

**Communicate**

I know it is the age or permissive marketing and spam, but it’s OK to send emails, especially if someone downloaded your product.   Pick up the phone and talk to those who respond via email or on your site.  Ask for a live chat via Skype or GoogleTalk with a customer who is having an issue.  Don’t worry as much about the finesse as the act itself.  Customers love to talk about their own experiences and not enough vendors reach out in friendly consultative ways.  After all, you are trying to improve the experience of using your product; not trying to necessarily close a deal.