The Developer's Guide to Speaking Business

You've undoubtedly encountered businesspeople in your corporation (be it managers, owners, executives, et al.) who truly have no idea what your work entails. They make what to them seems a simple request, you try to explain why it's *not* a simple request, and chaos ensues. They just don't understand, and you don't understand why they don't understand.

On the other hand, consider this: you've been working on a pretty big project and receive an email from a businessperson informing you that someone has opened the kimono on the whole deal and they're counting on you for some major blue sky thinking to run up the flagpole. This businessperson says you're going to have to find a way to move the needle by empowering yourself past the swim lane.

If you understood any of that, you're either being held captive by a startup or you secretly majored in business.

That email is an example of jargon, and it's part of the reason businesspeople have a difficult time understanding engineers. In fact, jargon—the term given to industry-specific language--has the potential to completely derail communication between fields. Jargon goes both ways, though: you can't start talking to a businessperson about entities or dependency injection or a disgruntled former employee's refactoring unless you enjoy the blank stare of an executive who's started to question your sanity.

Learning how to communicate with a businessperson makes your job easier, makes for a happier team, and can eventually lead to a better end result, so it's in everyone's best interest for you to bridge the communication divide between business and software engineering. Here's how to start:

1. **Never use acronyms.** There's nothing that could go wrong by *not* using acronyms, but there are infinite ambiguities that *could* go wrong if you do. For instance, you could start explaining that the OOP is taking longer to integrate than expected, and your businessperson could assume you were talking about something being Out of Print and that you're really just wasting time waiting for your collectible Rolling Stones album to arrive. Ambiguities love acronyms, so avoid them even if it's habit. Acronyms creep into speech in a misguided attempt to save time; in reality, using them wastes time because you'll just end up having to explain later. Avoid acronyms the same way you'd avoid using elevated speech or slang when talking with a non-native English speaker.
2. **"Just" is code for "I have no idea what it is you do here."** Whenever a businessperson says "just" to a software developer—as in, "I JUST need a button on the website that processes the order"— it's a sign that he or she wants results (usually within some ridiculously short timeframe) but doesn't understand the process for achieving those results. Your businessperson could, for instance, say on Friday morning that he "just" wants a new CRM by Saturday evening, but he wants a widget, not a functioning system, and doesn't care what sort of code

wizardry would be required to "slap it together." This is where being clear and assertive is in a developer's best interest: you need to explain to your businessperson you'll be happy to get results quickly, but that in order for those results to be sustainable, you'll need to write code correctly *the first time* and that isn't something that, out of respect to the company's future, you can rush. But what if the businessperson needs more of an explanation than that? Use our next tip.

3. **Analogies are your friend.** There are reasons the best orators in history used analogies to get their points across: they're a form of translation. When dealing with competing jargons, simple analogies are the best way to be understood by all parties involved. Top companies have mastered this concept in their marketing techniques and the most effective salespeople have as well. Go into any Apple store and ask for an explanation of the iCloud: you won't be given a breakdown of server processes. An Apple employee will most likely say, "Well, pretend that your iPhone is a house. You can put a lot of stuff in that house, but once it's full, it's full. The iCloud is like a shed in the backyard: it gives you more space to put things, but it doesn't change the size of the house." Analogies eliminate jargon and simplify complex concepts without insulting anyone's intelligence. Keep them simple and keep them relevant and you'll have achieved impactful communication.

4. **Frustration stems from a lack of communication.** If your businessperson is becoming angry or you're losing patience, it's probably because you aren't understanding one another. It's time for you to take a step back, calm down, and explain your process without using technical terms. If you're used to under-promising and over-delivering from a technical standpoint, take that approach to communication as well. Provide regular updates, explanation of processes, and sneak peeks into the work going into the code. You'll set yourself apart while also establishing trust.

Remember that communication with businesspeople is a lot like visiting Japan and expecting everyone to speak Spanish: you're going to have to change your approach if you want to get anywhere. Learning how to communicate with all kinds of people is the hallmark of a master developer.