Anatomy of an Interaction

Now that we've covered the main customer service techniques, we're going to dive into some of the nitty-gritty by looking at the anatomy of an interaction. These apply to any channel of IT support, email, phone, chat, or in-person interactions. From the first moment you interact with someone, it's important to think about how you say, "Hello." Do you make sure to tell them your name? Do you incorporate information you know about them in your greeting? Do you ensure a positive tone? Are your spelling your grammar on point? These are all ways to create a really good start to the interaction. Some of these things are hard to achieve though. I'm a horrible speller, especially when I'm in a hurry. But knowing some of these trouble spots ahead of time will let you find ways to address them before the interaction. For me, I know that when I'm in a hurry, I need to recheck my spelling before hitting "Send". Have you ever heard the phrase 'first impressions last a lifetime'? Well, that might be a bit of an exaggeration. It touches on some truth. How you first interact with someone will influence how the rest of the interaction plays out. I'm not saying you have to be over the top, gushy and nice, that might have the opposite effect. Just be professional, acknowledge the user, and show them some respect. Taking the time to get the interaction off to a good start will make everything that comes after easier. Let's check out two scenarios to see how this plays out.

bad:

Hi, Gail. How are you doing?

Not great. It's been a bad day trying to get my phone fixed.

What's wrong with your phone?

good:

Hi, Gail. How are you today?

Not great. It's been a bad day trying to get my phone fixed.

Sorry to hear that. Let's see what we can do to turn that around.

Just by acknowledging their feelings and demonstrate your desire to help them, you started to build a relationship with the user. Of course, you have to keep up the good work throughout the interaction, but laying the groundwork is an important first step. I remember that while you might have 100 issues in the ticket queue that need your attention, this is the only one that matters to the user. Show them it's your priority too. The next critical step in an interaction is how you respond to the user's questions. If they're taking the time to explain to you what happened, but

you brush off their concerns by acting uninteresting, things are going to go south fast. Remember to integrate the information you've been given into your conversation. This will show you're actively listening and can help them feel more connected to the interaction. Let's look at an example. Which one of these greetings do you think is the most effective? Greeting 1, "Hi, Rhori. How are you today? What can I help you with?" Or greeting number 2, "Hi, Rhori. I hope you're having a good day despite your computer randomly turning off. Let's see what we can do to fix your issue." It's important to be transparent with the user. If they start asking you a bunch of questions while you're still troubleshooting, you can do two things. First option, you can ignore them because they're just talking out loud. Second option, you can pause and say something like, "I'd be happy to answer all of your questions, but I want to look up this one first. I've written them all down though, so I won't forget them." If you say that make sure to write the questions down. To really build a rapport, try to remember a personal fact they've mentioned and bring it up later. Maybe they mentioned they love cats, later while you're waiting for something to load, ask them if they have any cats or what their cat's name is. This shouldn't be forced. If you're not the type to engage in small talk, skip it. Now, you're getting to the point where you're ready to troubleshoot, just make sure you clarify the person's issue before you start to troubleshoot. If you don't, you might find yourself going down a rabbit hole. Imagine that a user tells you their computer can't get online. You look at the IP address, DNS configuration, and you start pinging things with no luck, everything seems fine. Then 20 minutes later, you find out their machine is online, they just can't access a particular page. Had you clarified this at the start, you would have saved yourself and the user 20 minutes. It seems simple to clarify the problem space, but it's often overlooked. Take this example. Thank you for calling. This is Leon. How can I help you? Hi, Leon. My computer isn't working. That doesn't sound fun. What do you mean by the computer isn't working? It won't connect to the internet. Do you have the corporate password for the wifi? No, why do I need that? In order to connect to the wifi in the building, you need to use the corporate password. Well, I'm not in the building. I'm at a cafe.

That's odd. Your computer seems to be different than what we normally use. Can I get your name so I can look up your configuration? Ling Chan. Do you work at this company? No, my friend gave me the number. Problem-solving is a super important aspect of an IT support interaction. Being an IT support specialist means that you could be asked about anything. Even though you aren't expected to know the answer off the top of your head, you should know where to start looking to find it. People are coming to you because they have a problem they can't fix themselves. Sometimes they feel self-conscious about asking for help. Be aware of how you probe for information. Pummeling the user with question after question will probably create frustration on both sides. Make sure to set contexts and explain why you're asking the question. Saying something simple like, "In order for me to figure out what's really going on, I need to ask you some question," can make all the difference. When you're in person, things are a lot easier because you can see each other and read each other's expressions. But you might find yourself too comfortable. Imagine you're asking for help with your phone. You wouldn't want the person helping you to just take it out of your hands without asking. Make sure you tell the user what you're doing before you do it. If you're supporting a user remotely and need them to run some commands, don't forget to tell them why you need them to execute the commands. There's no need to go into a ton of detail. Without some context, you could strain the trust you've built. Make sure that when you're asking these questions or asking the user to run a command, you're really listening to the response, those little nuggets of information may help solve the issue. The last five minutes of the interaction will set the tone for how the user feels walking away from the interaction. Make sure to end on a

positive note. You might have solved their issue, but if they don't feel it was resolved or they're unsure of the next steps, then they're going to walk away feeling like it was a poor solution. How do you make a good final impression? Simple, reiterate the resolution, state the next steps, then ask the user if they have any questions.

Revision #1
Created 2023-05-29 14:55:07 UTC by naruzkurai
Updated 2023-05-29 16:41:21 UTC by naruzkurai