Stanford eCorner

The Five Whys

30-09-2009

URL: https://stvp.stanford.edu/clips/the-five-whys

Understand and question deeply the root causes – i.e. the "human problems" – behind every technical mishap, else your startup will be hindered in its progress. So advises entrepreneur and author Eric Ries, who suggests a layered analysis of decisions and procedures behind every technical undertaking. Follow up these questions with a proportional investment in human resources to alleviate further technical problems.

Transcript

And the last, I'll just go through briefly, is something called Five Why's.. The idea here is that whenever something goes wrong, we want to understand what's the root cause.. And this is a little bit frustrating, especially for the more engineering types ,because it means that behind every supposedly technical problem, there's actually a human problem that caused it.. And if you don't find out what those human problems are, you're never really going to learn and make progress.. So I'll just give you one quick example: Some code gets checked in and it takes the site down.. And you're like, huh, well, why did that happen? Some engineer used this really obscure API that if it's not used properly, it takes the site down.. It's like, that's interesting.. Why did they do that? Well, they're a new engineer and they were never properly trained in the use of this API.. Well, that's odd.. How come they were never properly trained? Oh, it's because their manager doesn't believe in training..

What? We thought we had a technical problem with our server going down and we actually have a managerial problem that relates to training.. My suggestion is that you do that analysis whenever something goes wrong, and then you make what we call a proportional investment at each of the five layers of the analysis.. So you don't say, "Oh, we found a problem; therefore, we're taking the next six months off to do prevention." But you also don't do nothing.. You try to find at least one improvement you could make at each of those five stages.. So we'll bring the servers back up, we'll fix that API, we'll go train that engineer.. Of course, we'll have a conversation with that manager.. But the manager might say, "Oh, sure, I'm happy to do a new training program for new engineers.. You know, that will take about eight weeks.. So, you know, if you don't want me to do anything else for the next eight weeks, I'll work on training," which is basically a fancy way of saying, "Screw you, we're not doing it." All right.. That's manager speak for not going to do it..

And the idea of Five Why's is this, instead of having that negotiation that's all or nothing, you say, "That's fine." I want you to do the first hour of your eight-week training program.. It's like, "What? I can't do anything in an hour." You say, "That's not true.. You could set up a training wiki and just create a page that says training program goes here.. And that's it.. That's all we're going to do this time.. But, see, the next time we have a training problem and we do Five Why's, we're going to notice this problem again and we're going to do another hour's work on it.. And that person will say we'll all create the training wiki.. It's like, sorry, dude, that's already done.. Now, you need to do hour two.. And if this problem keeps happening, we will naturally invest time wherever the problems really are in the kind of technical human combination system that is our product development team..

That's Five Why's...