亚麻官方STAR Example: Tell me about a time when you had to work with incomplete data or information.

My team had a big outage a few months back and we knew that there had been an issue because I received an alarm, I was on call. So I responded to the alarm, started looking at some of our dashboards and saw that there was a big drop in our order count. That usually signifies that something is gone pretty South for the customers. But the problem is I didn't have enough time to figure out what had gone wrong at a root cause level.

I knew I couldn't solve this just by myself because our system is quite large. And so I set up a conference call with several of the other engineers in the team and started a divide and conquer process and nominated a bunch of folks to start looking into different aspects of the system and start triaging.

Once we hit about hour four, even though we didn't know the exact impact on the customer, I knew that it was too risky to continue leaving this feature live in production. And so I made the call to do a rollback.

(你怎么知道这时候需要rollback而不是继续whatever it was before?
注意了！customer obsession 正确解题姿势：
There is a point of diminishing returns with an investigation like this and given the fact that this is a feature that any customer on Amazon can use, you'd never want to have sustained customer impact like this and we'd already gotten a couple of customer service calls. And so once you have a couple of confirmed customer service cases, it's kind of a good best practice to do a rollback and then allow for time to go and do the root cause analysis.)

It was a couple of days of detailed investigation, not only what the actual customer impact was finding out, you know, the number of customers impacted by the dollars that came out of that. But also trying to root cause and what subsystem we had a break such that we couldn't ship the orders that we needed to ship and once we determined what that was, we came up with an action plan to mitigate it. I was responsible for getting all the partner teams that were involved on board and making sure that they made space to go and deploy a hotfix and then getting that deployed and tested out in production and then finally doing deployment again.

Because we were able to root cause that we haven't had any issues since then. It also allowed us to expand our regression test suite. We prevented additional issues from happening in the future.

We also took steps to start monitoring additional services that will help us know there's something going wrong in that subsystem in the future.