Whenever there is a coordination problem, somebody says, "Hey, let's create a process!"

devblogs.microsoft.com/oldnewthing/20070905-00

September 5, 2007

Raymond Chen

Whenever there is a coordination problem, somebody says, “Hey, let’s create a process.” Now you have two coordination problems. I see this over and over, and paradoxically the people who create a process for managing a coordinating problem come off looking like proactive problem-solvers who get ahead of the problem. They’re the go-getters, the people who look at each problem as an opportunity for continuous improvement. All that great management buzzword stuff. It doesn’t matter whether or not the process actually works, because failure is an orphan, and besides, nobody follows up a year later to see whether your process actually improved things or whether it was abandoned six weeks after roll-out. You get the credit for proposing the process in the first place. Consider this problem at a hypothetical toy company: Some rumors have started that one of their toy cars is dangerous because of a wiring flaw that can result in excessive heat and possible fire. (It doesn’t matter what the crisis is. I just made up something vaguely plausible.) Employees all over the company are getting email from their relatives asking about the toys, but the employees don’t know what the story is either. Presumably, the safety department is investigating the problem and providing guidance to the PR and marketing departments so they can prepare a public response, and if there really is a problem, they’re working with the product recall team to organize how the recall will be carried out, and they’re working with the product engineering team to figure out what changes need to be made to the product to avoid the problem. In other words, the safety department is up to their ears in crisis. A employee from the dolls department, whose brother asked him about this toy car problem, can’t find any information on the status of this issue and sends email to the “Employee Issues” alias, and a dozen people chime in with “Yes, please, we would like this information, I want to know what I can tell my brother and everybody else who is asking me about it. I might even post it on my blog so I can try to counteract some of the out-of-control rumors that are out there.” A half hour later, somebody from the safety department responds, “Yes, we know about this, we’re working on it. Please don’t say anything publicly since we’re still investigating.” Somebody watching the saga unfold proposes a process by which employees can be kept up to date on the status of these types of emergent crises.

- Create an internal web site that contains a list of all current emergencies at the company, their status, what information is speculation and what information is solid, which pieces of that information can be revealed to the public, and recommended phrasing for the information when it is revealed.
- Whenever there is an emergency, the people responsible for responding to the emergency update the web site with the status of the investigation and the other information listed above.

This is the wrong solution to the problem. The reason why this is the wrong solution is that it suffers from the classic "Create work for somebody else" problem. The people who benefit from this site are not the same people who would make the site useful. This doesn't give the people who make the site useful much incentive to do so. They're busy dealing with the emergency. They don't want to waste the time of one of their valuable investigators to do "internal PR". This scenario repeats itself on a less dramatic scale with various types of inter-group coordination problems. One group wants to be kept advised of the progress of another group for whatever reason. But if the second group doesn't need the first group for anything, they have no incentive to keep the first group informed. For example, suppose the first team provides an interface that the second team uses, and the first team decides that they need to redesign the interface for whatever reason. During the transition, the first team provides both interfaces so that the second team can transition from the old to the new, with the request that the second team let them know when they have finished transitioning to the new interface so they can remove support for the old one. Unless the first team keeps on top of things, the second team will probably forget to tell the first team when they have completed the transition, or they may even forget to transition to the new interface at all! After all, the old interface still works.

In this case, the traditional way of making the second team care is to file a bug against them saying, "Stop using the old interface." That way, it shows up in their bug statistics as a reminder, and when they actually do the work, they can resolve the bug back to the first team. But you can imagine a scenario where the information the first team wants is more of a continuous nature rather than a single event. (Since bugs track single events.)

Raymond Chen

Follow