Le Chatelier’s principle in action: Notifications

devblogs.microsoft.com/oldnewthing/20180911-00

September 11, 2018

Raymond Chen

Some time ago, a configuration change was made to the Windows repo that caused some people (such as me) to receive hundreds of notifications a day related to a process known as the *git train*.

The git train is a process which merges changes between branches. It gets its name from the concept of the train.

Anyway, each time the git train did something, which it does many many times a day, I got an email notification from Visual Studio Team Services.

It took some work but I finally found the right people to ask to stop sending me these notifications. They promised to look into the problem and figure out what happened that caused the system to start generating tons of email notifications.

I suspected that most people in my situation had already created a custom email rule to auto-delete these messages, leading to another demonstration of Le Chatelier’s Principle for complex systems: “Every complex system resists its proper functioning.”

Some computer system somewhere is expending a lot of time and energy generating all the email notifications. This causes a bunch of other computer systems to expend a lot of time and energy to deliver those email notifications. And then the recipient’s computer system expends a lot of time and energy to delete those email notifications.

In total, an awful lot of time and energy was spent to accomplish nothing. It’s just two computer systems sending messages to each other, with none of the messages ever being seen by a human being at any point.

I resisted the urge to do the same thing and create my own custom rule to auto-delete the messages. I allowed the messages to arrive in my inbox, whereupon I deleted them. This served as a reminder to me to pester the people in charge to get them to fix their problem. If everybody set up a rule to auto-delete the message, the problem would never get fixed.

Raymond Chen

Follow