Every business process secretly wants to fail

devblogs.microsoft.com/oldnewthing/20211123-05

November 23, 2021

Raymond Chen

Business processes ensure uniformity and help catch errors. The people who created the process did so with the most noble of intentions, but secretly, every process subconsciously wants to fail. It just doesn't know it yet.

I got a bug saying that my widget does not meet XYZ requirements, and it contains a link on how to fix it. The instructions tell me to visit the XYZ dashboard on an internal Web site and gives me step-by-step instructions on what to do to fix the problem.

Step 10 tells me to click a button that is disabled.

There is a message next to the button that says, "Your widget does not meet the requirements for this button. Review this link for information on how to fix this issue." The link takes me back to the instructions. The one that told me to click the disabled button.

So I've come full circle.

The page also says "Problems found: Widget dcd0ae46937a has a doodad without an impact statement." (Repeat for three more widgets.) Those widgets are not my widget. I don't know why it cares about those widgets. There is also no way to search for those widgets.

It occurs to me that I can find widget dcd0ae46937a by URL hacking. It has one doodad, and there is an impact statement for it.

Now stuck, I email the team that manages the Web page for help. One of the team members replies, "Oh, I checked (unrelated web site) and found that widget dcd0ae46937a inherited doodads from its parent, and it's those doodads that don't have impact statements."

Okay, I contact owner of the parent widget, and they fill out impact statements for the inherited doodads.

I go back to the XYZ dashboard, and my widget has disappeared!

I ask the team what happened to my widget. No answer, but after a few days, it reappears, so must have been some sort of glitch.

I go through the instructions again, and once again, I get stuck at step 10 because the button is still disabled.

I URL hack my way into the other three widgets. I enter an impact statement for the doodad. I click “Save changes”. I get spinning dots.

I wait ten minutes. The dots are still spinning.

I open a new tab, go to the page. Oh, the impact statement was saved okay. It’s just that the “Save” button is broken.

All right, now that the doodad is fixed, I go back to my widget. Finally, the button is enabled, so I click it. I am asked whether my widget conforms to “Contoso quality guidelines”.

There is no link to the Contoso quality guidelines.

I ask the team where I can find the Contoso quality guidelines. They send me a link to the original page of instructions.

I’ve come full circle a *second* time.

I tell them, “Yes, I followed all those steps. I had to, because if I didn’t, the button is disabled. So why is it asking me whether I did something that the page already validates on its own?”

They say, “Oh, maybe that’s not the Contoso quality guidelines after all. Try this page.” They send me a link to the “Fabrikam quality guidelines. Fabrikam is a subsidiary of Contoso, or the parent company of Contoso, or something like that.

The Fabrikam quality guidelines document says, “To check if your widget meets quality guidelines, search for it in the quality report on web site Q.” Web site Q doesn’t exist. I find that web site Q was decommissioned and replace with web site Q2. Web site Q2 also doesn’t exist. Because Q2 was decommissioned and replaced with Web site Q3. Web site Q3 does not have quality reports.

The quality guidelines say, “You can also look at the Maelstrom report on quality guidelines, on web site Q.” Okay, so another link to the twice-dead web site. And of course, Q3 doesn’t have the Maelstrom report either.

Okay, well the next section of the quality guidelines show how to perform the quality check manually. So now I’m going through all the manual steps. Finally got to step 12 of 15. This says, “The widget must be registered in the widget catalog. To check if it is registered, please check web site W.” Web site W also does not exist.

So that’s where I am. We’ll see what the XYZ team says for this.

It’s taken a month to get this far, and I still cannot get my widget to meet XYZ requirements.

Every process secretly wishes to fail.

The kicker is that I don't even care about this widget! Another team created the widget, and they are the ones who use the doodads in the widget, but ownership of the widget got transferred to me because I took over the component that contains the widget. So I'm on the hook for getting the widget to meet XYZ requirements, even though I personally would be fine with just deleting the widget.

Epilogue: The team that created the widget originally did some investigation and concluded that they aren't using it after all. They told me that it's okay to delete it.

Bonus chatter: That every business process secretly wants to fail is another example of Le Chatelier's Principle for complex systems: Every complex system resists its proper functioning. I learned this and many other valuable lessons from the book *Systemantics*, which I wholeheartedly endorse.

Previously: Le Chatelier's Principle as applied to announcements and administrative overrides.

Raymond Chen

Follow