dcgross.com /fixing-ux-papercuts/

Fixing UX Papercuts

Daniel

3-4 minutes

How frequently do you find yourself mildly annoyed at software dysfunction? Tapping phone numbers in Google Calendar opens them in Maps. The Mac WiFi menu is slow. Google Docs doesn't let you share external docs on mobile. Etc. Etc. Lots of small papercuts.

Low Energy

Do you ever report these bugs? I don't. When I use something I'm usually trying to *get a thing done*. I'd be willing to, but I budget very little energy for this side-quest.

For years I've had my email plastered on my Twitter bio. Recently I opened my DMs. Messages started flooding in. Where were these people? Did they not see the email address in the bio?

I think the lesson from both of these points is that product usage follows a certain momentum and flow. Users might be willing to break momentum, but only slightly. You're on Twitter. The way to message people on Twitter is through DMs. Not email. If there's no DM option, well, whatever. Move on. Similarly, the user who experiences a bug with Google Docs isn't going to go through the trouble to write a detailed report to Sundar. Too much energy, breaking from the flow of usage.

Energy Efficient Feedback

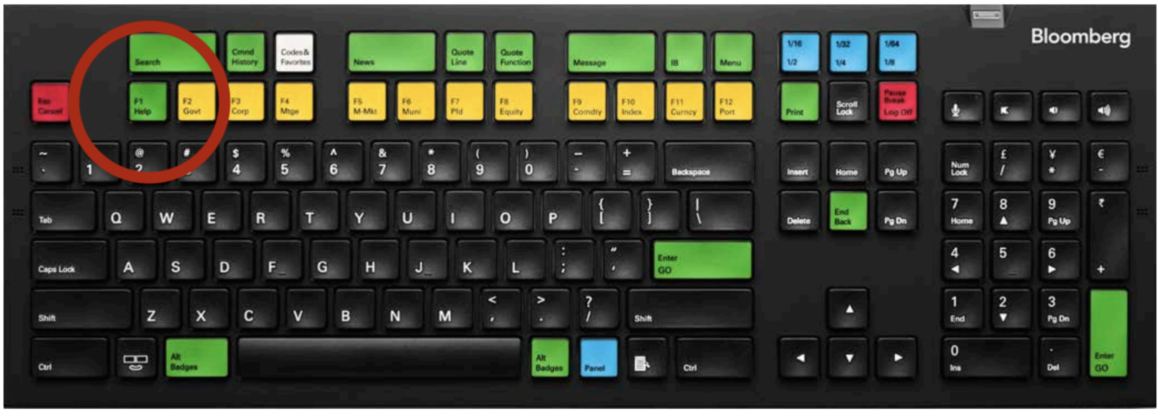
So how do you get feedback about your papercuts if nobody cares to report them? Two options.

1. Playtesting

Video games are a great place to study UX design, because the product *is* flow. Not sending emails. Not writing documents. Flow. You're *in the zone* when you play a good game. Even the smallest micro cut breaks that. PlayTestCloud is a popular tool for game designers. It lets you upload an APK, and instructs their testing audience to *narrate their thoughts* as they play the game.

Microsoft's RITE method is another example of this. I can't recommend these strategies enough. The thing I'm about to suggest next will seem more exciting to hack on, but still – nothing will beat the above.

2. Changing Equations



This is a Bloomberg Terminal keyboard. Custom built, it includes a dedicated Help button highlighted above. Tap it twice, and you'll get launched into an instant IM or phone call with someone at Bloomberg. 24/7.

This is good, but I think there's room to innovate further. I'd consider dedicating an entire block of your UI as an always-on textbox. I wouldn't make it a button like Intercom. A textbox. Every little bit of friction matters! It should be in a prominent place, and it should never go away. Let the user know they can type anything in there and you'll respond. Here's the twist:

When you do implement fixes, *announce and celebrate them*. Use Twitter, Instagram, whatever. Thank your users. What you're doing here is both decreasing cost (fluid, always-on UI) and increasing the reward (social affirmation) required in order to give feedback.

Hopefully these strategies help you identify your papercuts. I can't recommend the first approach more. Talk to your users and get them to spool the logs of their mind to you. Even small numbers will do.