



Home security-system SNAFU

A UX PUZZLE FOR THE BRAINY

Theme

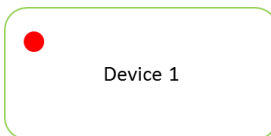
UX testing is great. We can prototype and test products to ensure they handle happy cases and certain edge cases in ways that make sense to users. This is primarily a product-centric activity.

When products are connected into system, as in IoT, unexpected behaviours may emerge. Unexpected behaviours are not limited to complex systems. This puzzle illustrates two dumb devices which together generate an emergent behaviour that causes multiple problems.

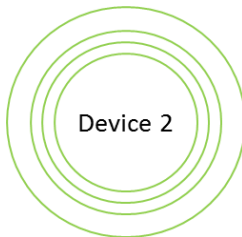
In addition to analysing this particular problem, we wonder what else we need to avoid problems like this. After all "Problems cannot be solved by the same level of thinking that created them¹."

Challenge

Your friends have bought two small home security devices.



This is a glass-break detector, part of their security company's offering. Its job is to detect the noise of a window being smashed. If it does, the light gives a blink. If the house is armed, the detector also sends a signal to the security company.



This is a standalone smoke detector. It beeps loudly if it detects smoke. It beeps once a minute when the battery is low.

For tidiness, your friends want to mount them next to each other in the kitchen, like this.



You tell them that this is a bad idea as there are unexpected consequences in doing this.

1. What are the unexpected consequences of this configuration?
2. In particular, who are the different types of user of the system, and how would they be confused?
3. Is it possible, even in principle, to eliminate this confusion?
4. What advice would you give them about arranging the devices?
5. Would conventional UX testing catch this type of problem?

¹ Paraphrasing "Albert Einstein"



Home security-system SNAFU

A UX PUZZLE FOR THE BRAINY

Pointers

Here are some pointers to help answer this challenge.

1. Create a diagram to show the inputs and outputs of each device. Look for unexpected coupling between the devices.
2. Identify the two recipients of messages from the system.
3. Would the type of UX testing that you do catch this problem? What additional skill set would you need?
4. What could the designers of the devices have done to prevent the problem from happening?

More Pointers

Here are some more pointers to help answer this challenge.

1. Installers of the glass-break detector test their installation by clapping their hands. If the light flashes, the detector is working OK.

We will give you some possible answers in the next post.

Good designing!

About the Puzzles

These [puzzles](#) show how, by taking the user's point of view, we can generate ideas and select the best ones, innovate, avoid pitfalls, and generally make better thought-out user experiences. They are a fun way of presenting serious ideas about designing information rich solutions. For a more serious and systematic way, see "Experiencing + Architecting Information" at www.theinformationartichoke.com.

To bring this type of training into your organization or educational institution, contact Martin at theinformationartichoke@gmail.com