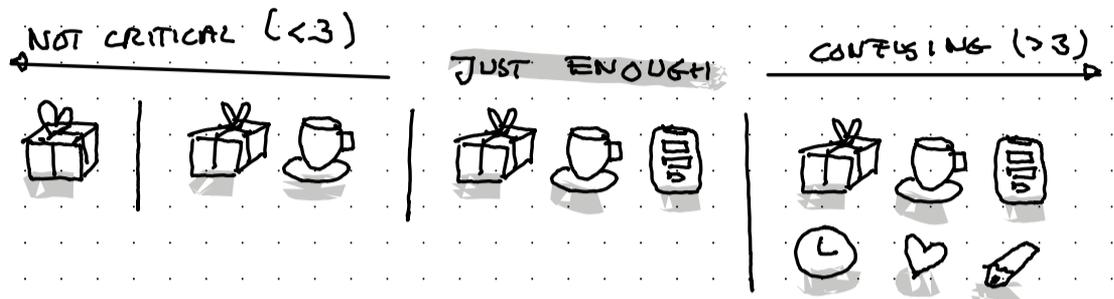


How many different prototypes should I build or test?

In short: I'm a sucker for building and testing very different ways of solving the same idea in order to learn the most possible. And three is my favorite number. It's enough to push people to imagine that there could be other ways of doing, enough to make them critical, and not too much so that it doesn't confuse people.



The rule of three

When it comes to prototype, I love to present three different things. When you show one, it's like showing a photo of your unique child. People can feel that this is a sacred thing. So it's hard to criticize.

Plus when you show one thing, people get in a feedback mode about the details. They don't think about what else could there be.

When you show two it's like a boxing match. It's either this, or that. Or it's like the last round of an American or French presidential election. You have to choose between the least worst candidates.

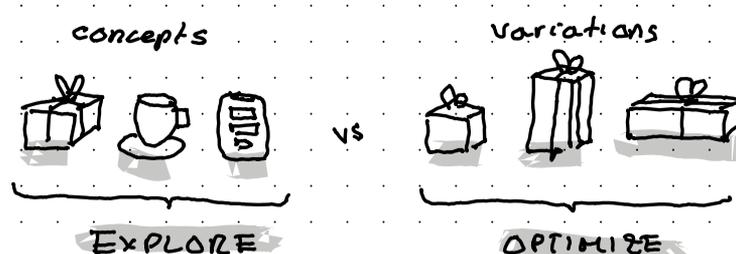
When you have three it makes clear that the decisions have not been made yet. That things are still open. And therefore people tend to play LEGO. They say: from the first one, I'd keep this, but remove that, and from the second one, I'd keep that but remove this, etc.

And they can more easily imagine and dream about something totally different that you didn't even have on your radar.

But why not more? I feel that once you are beyond three it gets messy for people's head. They have a hard time remembering what was the difference between number 7 and number 1. And it feels like too much choice, just like when you are in a new supermarket in a new country and you have to choose a yoghurt. There is just too many options to choose from, which might lead you to not even choose (what smart people call **the paradox of choice**).

Different concepts, not variations

When I say three prototypes, I mean three very different directions. Not tiny variations like it's usually done in AB testing, where you swap a color, or a text. AB testing has its place in the world of optimization. But it's not the approach that leads to the deepest learning moment.



PROTOTYPING vs AB TESTING