

## KEY QUESTIONS FOR LEADERS

These are questions we ask each other before, during, and sometimes after we work on something. That something can be as small as a couple-hour project or as big as something that takes a few weeks or more. Either way, it's important to ask questions like this in order to make sure you're doing work that matters.

- **Why are we doing this?** Ever find yourself working on something but you don't know why? Someone just told you to do this or that? It's pretty common I think. It's important to ask yourself (and others) why you're working on this. What is this for? Who benefits? What's the motivation behind it? Knowing the answers to these questions will help you better understand the work itself.
- **What problem are we solving?** What's the problem? Are customers confused? Are we confused? Is something not clear enough? Was something not possible before? What problem are we solving here? Sometimes when you ask yourself this question you'll find that you're solving an imaginary problem. That's when it's time to stop and reevaluate what the hell you're doing.
- **Is this actually useful?** Are we making something useful or are we just making something? It's easy to confuse enthusiasm with usefulness. Sometimes it's fine to play a bit and build something that's cool, but it's worth asking yourself if it's useful too. Cool wears off, useful never does.
- **Are we adding value?** Adding something is easy, adding value is harder. Is this thing I'm working on actually making the product more valuable for people? Can they get more out of it than they did before? There's a fine line between adding value and subtracting value. Sometimes adding is subtracting. Too much catsup can ruin the fries. Value is about balance.
- **Will this change behavior?** Developers have a tendency to add stats to a screen just because they can. Counts, totals, sums, averages. Numbers can look cool, but do they change behavior? Does it matter if someone knows there are 38 of these instead of 42? Does it matter that someone knows it took 0.08 seconds instead of 0.02? Sometimes it might, but it's important to constantly ask yourself: Will knowing this information change someone's behavior? Can they do something useful with this information? Will they make a better decision because of this information? If not, pull it out of the interface. Data without purpose is noise.
- **Is there an easier way?** There are lots of ways to do things, but for simplicity's sake let's say there are two primary ways: The easier way and the harder way. The easier way takes 1 unit of time. The harder way takes 10 units of time. Whenever you're working on the harder way you should ask yourself is there an easier way? You'll often find that the easier way is more than good enough for now. Most people's problems are pretty simple — we just imagine they are hard.
- **What's the opportunity cost?** What can't we do because we're doing this? This is especially important for smaller companies that are more resource constrained. Limited time makes prioritization more important. If we work on feature A can we still do Feature B and C before April? If not, would we rather have B and C instead of A? Is A really worth the opportunity cost? Ask this all the time.
- **Is it really worth it?** This one should come up all the time. Is what we're doing really worth it? Is this meeting worth pulling 6 people off their work for an hour? Is it worth pulling an all-nighter tonight or could we just finish it up tomorrow? Is it worth getting all stressed out over a press release from a competitor? Is it really worth spending \$1000/week on Google Ad words? Is it really worth...?