

## Raising an issue

All projects encounter challenges that must be discussed. It's important to think about your audience and desired outcome in raising an issue. The goal is not to play "hot potato" to whomever responds first, but rather to set the stage for your desired outcome — whatever it may be.

Questions to ask yourself:

1. Where should this issue be raised?
  - a. Is there anything sensitive about this problem? If so, it might be worth asking your tech lead, engineering manager, or product manager directly first. If not, bias towards asking in the appropriate (preferably public) project channel.
2. In addition to the decision maker(s), who else needs to be informed? Make sure they're in the email or channel you're posting to.

Components of a successful post:

1. Outline what the risk is.
2. Outline what happens if the risk occurs for example, user-visible errors, degradation of experience, server errors, or your product being taken down.
3. How likely is it that we will end up in any of the above situations (2.)
4. What is your recommended course of action if the above situations occur?
5. Who/what is affected by these risks?
  - a. Is the situation limited to a specific user population, such as the individuals working on the feature, dogfooders, or customers in production?
  - b. Is the situation limited to a specific launch milestone, such as an upcoming dogfood release?
6. Decision maker and call to action.
  - a. Who needs to make a decision on the course of action? Make sure to mention them explicitly. If you're not sure who needs to make a decision, call that out instead.

## Example posts:

- "Hey <product manager>, I'm worried about project X hitting our current timeline if we don't cut scope. Currently, we have no buffer between our projected code complete date and the code freeze to make the general availability (GA) launch timeline. I'd like to recommend we cut A,B, and C tasks which will bring up the buffer to five days; we could complete those tasks as fast-follows."
- "Currently, feature X involves calling our update API. We have only load-tested that API up to 10,000 calls per second. If we exceed that capacity, users will hit an indefinite loading spinner. <Tech lead>, is N an acceptable assumption for launch? <Product manager>, do we need a different error behavior?"
- "I'm not sure which product manager owns this decision, but we need to decide if we do X or Y in order to stay on track for our dogfood launch. <Product manager>, <engineering manager>, or <tech lead>, can you help me find the appropriate decision maker?"
- "<Engineering manager>, I'm worried I'm not going to finish X by Friday like we talked about. Unless we take me off bug triage for the rest of the week, I'm pretty unlikely to finish until Tuesday. Are you ok with that?"