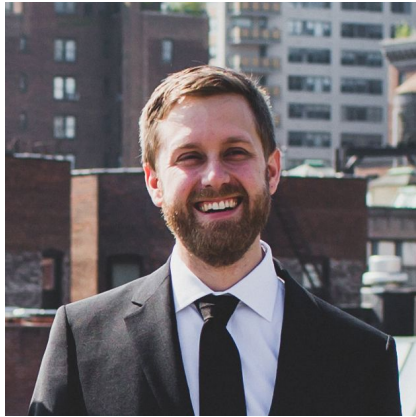


Title:

Building effective roadmaps

Author: Travis Donia

Headshot:



Effective roadmaps help a cross-functional organization make deliberate choices about a product's development. Whether you're writing a roadmap or reading one, you've got a stake in making sure the roadmap answers the questions you need stakeholders to provide consensus on. Here are some questions that good roadmaps answer.

What impact will this have on users?

Outcomes are more important than outputs. When building a product roadmap, it helps to be explicit about the types of behavior that you intend to encourage. In Agile development, this intent is codified in user stories: *As a* [type of user], *I want to* [activity]. In other development methodologies, use cases perform a similar function. Regardless of how it works in your organization, the goal is the same: put the user first and speak to the behavioral impact that you want to create, not just the outputs that must be built.

How will we measure impact?

It also helps to know how that impact will be measured—especially if your team will need to implement the metrics. Work with stakeholders to get consensus up front on how you'll measure and have them ballpark the impact they expect a feature to have. Document your hypotheses. Prior to implementation, those estimates will provide a gut check for the roadmap—is this outcome realistic? Post-release, comparing the hypotheses with actuals will help create a feedback loop that will improve your ability to estimate impact in future releases.

What are our strategic objectives?

As an engineering leader, understanding the strategic objectives that underlie a series of releases will help you guide your team on the abstractions that will be long lasting, the expectations the business holds on scale requirements, and where new interfaces will need to exist between teams and services. It's also a good way to talk about timing and the order of operations. Even if these objectives are documented elsewhere, explicitly referencing the OKRs you're building towards will help make sure your plans are consistent with your goals.

What are we willing to give up?

There's a wide range of features you *could* build; the roadmap is intended to capture those that you *will* build. Out of a broad product opportunity space, the roadmap articulates the specific choices that have been made. Sometimes, that strategy is most evident in the reflection of what you're willing to give up. To do this, consider the opportunity costs—the features you're not pursuing—as a way to make explicit the product values you are expressing. For example, if your market places a high value on privacy, there are a number of social features that you won't be able to pursue. By surfacing those trade offs, you can expose the strategy underneath them.

What are we testing and what mistakes are we comfortable making?

Another important function of a roadmap is to surface the places where there's uncertainty. Uncertainty is part of the process; if the right answer was obvious, someone else would have built it already. For your team, flagging the areas of the implementation under active testing is important for managing the level of attention you invest in an implementation.

Similarly, not every step will take you forward. If you knew you were making a mistake, you probably wouldn't make it. With that in mind, effective roadmaps should also flag irreversible steps and make sure there's consensus among stakeholders that the risk is worth it.

How will these features roll out?

Effective roadmaps help coordinate the release by indicating how features will be introduced to users. The marketing team will probably appreciate having visibility into the work they'll need capacity for. Assume the same is true for the other teams you work with; use the roadmap to identify those dependencies upfront.

By covering the plan for releasing features in the roadmap, it will help build alignment and de-risk a release. For instance, if you know that a feature has open questions and is being actively tested, you can recommend giving preview access to a small set of customers and collecting their feedback before turning it on for everyone. Anticipating this in the roadmap will make it significantly easier to implement feature flags during development and will provide the other teams you're working with the flexibility they can use to answer the open questions.

Good roadmaps create momentum as more teams' needs are represented. That doesn't happen by accident. Someone in the organization is working to make sure your product's roadmap spurs the coordination that makes the product better with every release. Even if you're

not responsible for crafting the roadmap, you can make it more effective by helping ensure it's answering the right questions.

Thank you to Stephanie Wai for reading and editing drafts of this essay.