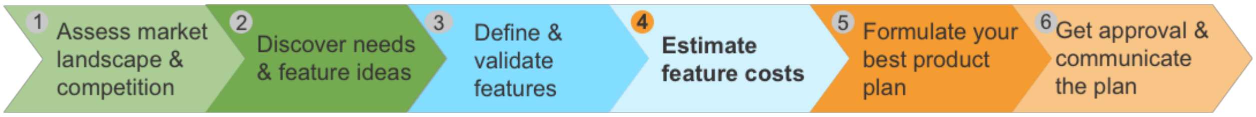
Plan 04: Estimate feature costs for product planning


1 Assess market landscape & competition
2 Discover needs & feature ideas
3 Define & validate features
4 Estimate feature costs
5 Formulate your best product plan
6 Get approval & communicate the plan

A great product plan includes the features that achieve your product objectives *and* that are feasible given resource and time constraints. But estimating the effort (cost) required to complete features is challenging, especially when you don't yet have complete clarity around what to build or how to build it.

Problem: It takes time to estimate the effort required to develop a feature, because first you have to understand its scope and feasibility. But you don't want to make a big investment in investigating it, if it's unlikely to make it into the product. You need a faster way to come up with a sufficiently accurate estimate that you can refine later in the process.

Solution: Before you estimate, the product manager should weed out lower-value feature ideas. Then focus on high-value features under consideration for the next product release or two, use a consistent method for estimating, and plan for risks and inexact or imprecise estimates.

Checklist

☑ Use a consistent method for estimating feature cost.

☑ Plan for risk and inaccurate estimates.

☑ Compare estimates to actuals so you can improve.

1. Use a consistent method for estimating feature costs.

Your product development team should lead this effort, because they own the resources that will build the product. The UX/UI design team should be involved, too, especially when the focus is on end-user experience. The product manager should be a key collaborator.

Use a consistent, documented method to estimate costs so that it's easier to compare features when you're making planning decisions, and your process is scalable and teachable. Best-practice methods for estimating feature costs in the planning process should:

- Enable you to estimate feature cost before you commit to build the feature. You probably won't have all of the requirements you need for a more accurate estimate.

- Document your assumptions.
- Include risk assessment for larger-scale, important features.

The "T-shirt sizing" method may not be enough. If not, here's another method:

Calculate quantitative time estimates using days, effort score, or another numeric system (like Agile planning's pebbles, rocks, and boulders—as long as your organization has specific, defined time estimates for them).

Consider these tasks and others, as needed, for your product:

- Engineering build time: front end, back end, API, systems, integration, etc.
- Spike and Design: research, discussion, and in some cases prototyping or doing a proof of concept exercise before the Design phase to validate assumptions and accuracy of intended results
- Iterative prototyping with user testing
- UX Design
- Code review by peer (pull request)
- Quality assurance (QA)
- Other work that is part of feature development for your product, such as documentation

Start by estimating engineering build time (the first bullet above) and round up. Then add about 30% to account for the other items listed above (in total, not for each).

For more information, see *Software Estimation: Demystifying the Black Art* (Steve McConnell, Microsoft Press) and *Estimates and Estimation*, Dave Ross, blog post June 8, 2017 on 10up.com. Just note that they contain more detail than you need in the Plan Phase.

2. Plan for risk and inaccurate estimates.

You know what they say about the best-laid plans. Things can and do go wrong: You lose an engineer you were counting on; you discover that your implementation approach isn't going to work; you uncover dependencies that cause delays; and more. But you don't want to over-pad your estimates to account for everything that could go awry.

Instead, document the risks, their likelihood of occurrence (low, medium, high), impact (what would happen), and mitigations (ways to reduce likelihood or reduce impact).

For example:

Risk	Likelihood	Impact	Mitigations
The only engineer who really knows this part of the product could leave the company	Low	Anyone else will take at least twice as long to complete the work.	Document that part of the product. Assign someone else to work with him/her as peer on this part of the code.
Our proposed implementation approach may introduce security issues	Medium	Until we get into detailed design, we can't be sure this approach will work. If not, we have to revisit and/or increase time estimate.	Spend time now investigating the approach. Build in extra time for the spike and address penetration testing results before going live.

Adjust your estimates by including mitigation efforts if the likelihood is medium or high.

What to do when things go worse than expected

Estimates tend to be optimistic. Features turn out to be more complicated than you originally thought, and unexpected issues can and do crop up—like a critical bug that sucks up engineering time.

When that happens, the product manager should work with engineering to identify alternatives and present them to stakeholders. They could:

- Postpone the feature to another release.
- Break down the feature into sub-features, one or more of which you can complete for this release, one or more that you postpone.
- Drop the feature and, if possible, replace it with the "next most desirable" feature that's available resources can complete for this release.

What to do when things go better than expected

Yes, it happens—all features in the release are done, and there's time to spare. Typically, development teams have a backlog of defects and tech debt to work on. But if there's time to add something to the product plan, deciding what to add is easier if you've already prioritized new features and have initial costs for them.

3. Compare estimates to actuals so you can improve.

Estimating accurately requires skill and practice, so include a review of estimated versus actual time as part of your release review process.

Answer and document these questions:

- If the estimate was off by more than 10%, why? Did we foresee, or could we have foreseen, why the estimate was off? What did we miss (retrospectively)?
- How can we do a better job of assessing and mitigating risks next time?
- How can the team improve estimates?

Next Best Practice: 05 – Formulate your best product plan

Put your best product forward™

Created by product people for product people, Obo is a product decision system that guides you to make the best decisions about what's in your product plans.

Call +1.650.926.9900 or email demo@obo.pm to schedule a demo now.