

# Are you Solving Customer Problems or Just Building Features?

[Nick Fields](#)

When your product can do more than it could do before, that sounds like a good thing. Added functionality, new capabilities, a more robust feature set... these are the talking points product marketers salivate over and executives search for on product roadmaps. But are you solving for actual customer problems?

In the never-ending race to ship out product updates to keep up with—or ideally, stay ahead of—the competition, it's easy to get caught up in the flurry of activity and become a [feature factory](#). The danger, however, lies in mistaking new functionality for actually adding meaningful value to the customer experience.

All those new features might look good on a product comparison matrix and give salespeople a new angle when pitching reluctant prospects, but none of it matters if those features aren't solving real customer problems.

## Where are product teams getting their feature ideas?

According to our [2023 State of Product Management Report](#), customer feature requests are still the top source of actionable product ideas, but that's only the case for 35% of respondents. Feedback from sales and support is the source 26% of the time, and the competition inspires new features for 16% of respondents. Most concerningly, 19% of respondents reported that their top source of ideas comes from senior management, who are often disconnected from both customers and the product development process.

To reframe things, only about one out of three feature ideas actually come directly from customers... you know, the people who are paying money to use your products. And while a quarter comes from sales and support, these too can come with their own caveats since there's always some amount of interpretation and bias involved.

When product investments don't meet expectations, the blame gets spread around. But a lack of a clear company strategy (23%), poor prioritization (19%), misallocation of resources (17%), and underdeveloped roadmaps (17%) encompass the vast majority of misfires. These all point to internal deficiencies in planning rather than problematic execution. In short, organizations are making the wrong bets rather than messing up on the go-to-market or implementation fronts.

## **Why do product teams become feature factories?**

No one intends to waste valuable resources on projects that don't yield meaningful results. No software developer wakes up in the morning excited to write a bunch of code that will be re-written a few sprints later. Product managers don't want to ship features no one will use. And no management team prefers an extra bullet point in a sales deck over another satisfied customer.

But despite good intentions, [feature factories](#) arise more often than anyone cares to admit. There's no single cause for these misguided projects frittering away precious development and testing resources, but there are some common causes for many of these missteps.

### **A scattershot strategy**

In the quest to meet the needs of as many different people and customers as possible, product teams can become too ambitious and reactive in their strategic planning. Saying no to a key customer or stakeholder is challenging, so the product team finds itself pursuing multiple paths

simultaneously.

Implementation teams get divided up to solve lots of different types of problems, often in the same sprints. This can incrementally add new capabilities to the product and potentially quell a few customer complaints, but without a unified vision, it can also lead to a lot of partial fixes that don't attack and solve major pain points in a holistic manner.

Instead of the product team putting more energy into validating and solving major problems, customers get a steady stream of tweaks and minor improvements across the board. This generates a sense of progress and improvement in the form of minor enhancements that may feel like you're on the right path, but never quite puts those issues to bed from a customer perspective. It's a bit like treading water. You might stay afloat, but you're burning a lot of energy without getting closer to the shore.

## Noisy customers

When it comes to your product roadmap, [the customer is most definitely NOT always right](#), especially when they ask for specific features. This doesn't mean customer ideas and requests should be ignored... quite the opposite! But it does mean product teams should ultimately be responsible for defining features.

When a customer has a feature idea, it's not really because they want that particular feature. In the majority of cases, they have a problem or pain point they want to address, and their suggested solution is their way of asking for help. But most of your customers probably aren't product managers. They just want the problem to go away and this is their best guess on how to solve it.

If you build exactly what the customer asks for, it doesn't guarantee they'll get what they actually want. For products to truly address the root cause, product managers must understand the jobs their customers are trying to

get done and identify the roadblocks preventing them from achieving it.

While it's certainly possible that the feature request is spot-on, chances are it doesn't fully address the underlying issue or will only address a very narrow set of use cases that don't necessarily fit in with your product vision.

By really digging into the problem space with customers and putting in the right validation efforts, the product team can get a clearer picture of the right solution opportunity. Then they can work with the implementation team to shape a more holistic solution that improves things for a broader swath of current and potential clients.

## **Shiny object syndrome**

When an executive with enough influence gets excited about something, their enthusiasm can sometimes create a runaway train. They could be inspired by new technology, a new business model gaining traction, or a competitor's differentiating capabilities. Whatever it is, these power brokers get it in their heads that THIS is the game-changing addition the product needs. This comes regardless of any existing customer problems to be solved.

They bypass ROI calculations, due diligence, prioritization exercises, and customer validation. They lobby hard for their [new obsession](#) to jump the queue and get added to the product. Fearing blowback or simply trying to play nice with this executive, product managers just go with the flow and disrupt their own roadmap and strategic plans to make it happen.

When it hits the market and doesn't move any needles, there's less scrutiny and criticism because who wants to tell the boss that their bad idea was a flop? And without some brave souls willing to take a stand, it's likely to keep happening as long as that power dynamic continues.

## **How can product teams focus on value versus**

## volume?

To ensure product development resources get expended on actual customer problems, product managers must dig deep into their toolbox. Relying on some of these fundamental principles will introduce more discipline into the process and give the product team solid footing to stand their ground and make some potentially unpopular decisions.

## Learn to say no

Nearly everything on a product roadmap is there because someone thought it was a great idea or asked for it. Unfortunately, there are way more great ideas and requests than there is time to address them. We can't do everything, so we must be choosy.

This puts product managers in the awkward position of having to [say "no"](#) to stakeholders and customers. It's awkward because plenty of us are people pleasers. Plus it seems odd to say we're trying to satisfy customers while regularly telling them "we're not going to give you what you are asking for."

Product managers don't do themselves any favors when they agree to do everything asked of them. It's not possible to deliver on those promises and it's an irresponsible use of resources to say "yes" to everything.

Product management would be a pretty easy job if all we had to do was take requests and hand them over to developers. Deciphering which requests will drive [key results](#) is the secret sauce. Luckily, saying no is a little easier when there's an objective prioritization process guiding those decisions.

## Use prioritization frameworks

Every decision to build a new feature is a judgment call. But making the decision on any one feature in a vacuum presents problems. There's no context nor much consideration given to what DOESN'T get built instead.

Using prioritization frameworks eliminates these isolated judgment calls. They force everyone involved to both justify the need for a given feature and stare at the trade-offs head-on. Consider selecting a framework that incorporates scoring and ROI. These are particularly helpful for avoiding prioritizing features that don't move the needle in meaningful ways for customers or the business.

The [opportunity scoring framework](#) puts customers in the driver's seat, which should lead to prioritizing features that customers actually care about. The [jobs-to-be-done framework](#) is another tool that refocuses prioritization around what customers are trying to do rather than what the product team thinks they need.

There are [many prioritization frameworks to choose from](#), so teams should try out a few to find their best fit. You don't have to pick the perfect framework. The most important thing is to use ANY of them (if not [multiple frameworks](#)). These are tools that can help force more objective conversations. Plus, they help you directly acknowledge the tough trade-off decisions that need to be made.

## **Validate with multiple customers**

It's natural to want to make customers happy. We are in the "delight" business after all. But the problem facing a particular customer may or may not represent a true systemic need. It might be unique to that customer due to external factors or just not be a top priority for others.

The only way to fully comprehend the scope of the issue is to engage more customers. Discover whether this is a real problem for them as well. If it's a common complaint causing customers a significant level of pain, then it likely warrants prioritization. Otherwise, there are likely better things to work on.

Some enterprise clients may still get their personal wish list items fulfilled.

But the organization should go into things with their eyes wide open regarding the eventual ROI and impact of dedicating resources to a relatively bespoke situation.

## **Define KPIs and success metrics upfront**

Don't build something, release it, and then figure out how to gauge its success. Rather, product teams must begin by defining measurable indicators of the outcome they're aiming for.

If the team can't come up with any solid way of assessing the ROI or impact of a new feature, then it's time to revisit the rationale altogether. No net-new features should make it onto the product roadmap without a measurable benefit for the business/customer base.

## **Adopt a theme-based approach to roadmapping**

The antithesis of scattershot feature releases is [leveraging themes](#) to dictate product development. Themes create an overarching thrust for an entire development cycle.

This enables the product team to prioritize multiple features to be worked on at once with the ultimate goal of building out significant functionality to comprehensively solve a specific problem area. Instead of making parts of the product slightly better, it takes one problem and tackles it head-on.

This singular focus yields benefits for the entire business. Product teams can do their homework and fully understand a problem, providing valuable customer insights to the organization. Implementation teams can dedicate more resources and problem-solving energy toward identifying comprehensive solutions. Sales and marketing have a meaty set of capabilities to sink their teeth into for go-to-market and customer communication purposes. And senior leadership can get out of the weeds and focus on big-picture priorities.

# Keeping customer problems at the forefront

Resisting the pressure to continually push new features instead of focusing on customer problems might give you a case of imposter syndrome. But rest assured you're doing the right thing. Customers aren't counting how many new bells and whistles your product adds. They just want it to do what they need it to do.

So use the strategies above to keep your product on track and your team focused on what really matters. To bolster yourself, [here's more on why you should put your customers first](#).