

8 tactics for better innovation

Times like these call for quickly rethinking how your team does things. So, how can you surface new options, learn fast, and find better approaches?



Try these tactics to experiment more systematically and effectively.

1. Focus on how your core users' needs are changing.

Understanding the lives and challenges of the people your team serves — whether they are internal or external customers — is the critical jumping-off point for imagining how to make things better for them. In times of great change, your core users' needs are likely changing, too. Your previous research and institutional knowledge about them might no longer apply or, at least, may need to be reconfigured.

To ensure that your team's work stays aligned with your users' evolving needs, start by asking your team for their observations using questions like:

- *"How have our end users' goals and pain points changed?"*
- *"What's most pressing for them right now — and of those things, which are most likely to remain important over time?"*
- *"What assumptions about them that we've built up over time are now questionable?"*
- *"What data do we have to support these new observations?"*

To supplement and validate your team’s observations of your core users, consider short surveys, a handful of interviews, and/or a focus group with them. [Research suggests](#) that even small samples of users (as few as three to five) can provide surprisingly helpful insights that allow you to make good decisions, especially if you gradually build on what you learn by checking with users again during your innovation process to be sure you’re on the right track.

Be sure to keep your findings front and center as ideas swirl and gain momentum. Your users’ needs can — and should — serve as a touchstone when debates arise over which potential changes are worth testing.

2. To spark ideas to test, look at what others have done.

Contrary to popular belief, breakthroughs are rarely 100 percent original. More often, they are small improvements on what’s already been tried. For example, Apple’s iPod wasn’t the first MP3 player — it was the one with the best design.

What existing concepts — within your organization and within and beyond your industry — can you and your team build on to better serve your users? To develop options, you can:

- **Reach out to peers who work in different departments of your organization (or in different organizations in your industry).** What changes have they made or are they considering making? And what are those “we already tried but they didn’t work” ideas that you might reshape to succeed in the new reality? Maybe another company’s sales team is bundling offerings in a way that you should be, too. Or maybe marketing’s self-service kiosk concept that fell flat last year is worth revisiting.
- **Draw inspiration from innovation success stories in different industries.** Musicians holding [concerts on video game platforms](#), financial companies adding new services to help clients apply for relief loans, [a corporate floral company](#) rethinking its customer base and offering virtual bouquets — there are countless successes stories to learn from in the current whirlwind of change.

3. Select ideas to test based on usefulness first — then feasibility.

Unless you have endless time and funds, you'll have to make some tough calls about which ideas are worth testing. Effective innovators often select ideas by considering how useful the proposed change will be for users and how feasible it is to implement, with a tendency to place greater value on usefulness: A really useful change that's hard to implement typically holds more promise than a feasible change that's not all that useful.

For example, let's say you manage a help desk team for a new piece of remote-work software, and email volume has recently doubled. Users are experiencing long help desk wait times, and you worry that if wait times persist, poor service could drive away customers.

To zero in on good options, list out all of your ideas and evaluate them by asking:

- **Which ideas seem *very useful to your end users*, but *not very feasible*?** Maybe it would be extremely useful to build contextual help or software wizards into the product so that users get step-by-step guidance during their sessions. But your team doesn't have the time, skills, or budget to make that happen. Still, don't dismiss the idea outright. Consider modifying the idea to become more doable or devising a long-term, cross-team plan.
- **Which ideas seem *feasible*, but *not very useful*?** Maybe it's easy to use the small budget you have to hire a contractor to help field users' emails. But that hire wouldn't know the intricacies of your product as well as your team does and would potentially drain team productivity for training and support. Meanwhile, users could become even more frustrated if their first interaction is with someone who couldn't help them quickly.
- **Which ideas seem *both useful and feasible*?** Maybe team members could write short instruction guides covering the most common issues users have with your software. Then they could email the appropriate guide to users who write in about that issue. That could lighten the load or serve as an interim fix while you work on other solutions.

If no clear best option emerges, could you break your team into groups and experiment with different options simultaneously until a "winner" emerges (as management expert Linda Hill describes Google doing in [this Ted Talk](#))? Or could you try quick, scaled-down experiments for each option?

4. Develop research questions to inform your test parameters.

You might have an educated guess about what will happen when you make a change, but be careful — those hunches can lure you too quickly into solution mode, causing you to overlook important factors and to end up with a solution that doesn't adequately address the problems. Instead, start with the fundamental questions you want to answer, which will help you design a more effective test.

For example, imagine you work for a hotel that uses entry key cards. Your team is rethinking the check-in process to improve sanitation for staff and guest safety and peace of mind. You don't have the budget to switch to a keyless entry system, so you're considering whether to purchase UV lamps to disinfect key cards.

Your research questions might be:

- How will we know if the UV lamps help safeguard staff and guests — could we compare pre- and post-lamp sick days among staff, for example?
- How will we know if the lamps alleviate guests' anxiety — could we add a question to the email surveys we send out to guests when their stay is over?
- How long should we track these data points?
- How big of a difference do we need to see in these metrics to know that the lamps are worth buying?
- What procedural obstacles might come with the UV lamps — how much time will using the lamps add to check-in wait times?
- Could we compare the UV lamp metrics with those from another possible solution, such as gloved staff wiping key cards with disinfectant in front of guests?

5. Give users a rough or partial version of your idea to react to — not a finished product.

One of the biggest questions teams face when they experiment is how much time and money to invest in building out an idea before testing it. Slapdash prototypes can yield suspect results; your users might not be able to adequately experience or visualize the idea you're testing. But full builds are expensive and invite [confirmation bias](#); you've invested too much to interpret the results objectively.

Aim for something between these two extremes. Choose an approach that works for your industry and your team's function. Some options:

- **Prototypes and partial builds.** Companies use a huge range of options, from renderings of an idea or potential product to digitally printed early versions to guerrilla selling (e.g., sending out email offers for a service that doesn't exist yet, then offering gift cards to people who want it). One real-life example of a partial build: To test whether people would buy shoes online, [Zappos started](#) by building a website, then buying shoes from brick-and-mortar stores to fulfill orders. They added their own inventory only after they knew the idea would work.
- **Phased launches.** For example, a restaurant might start doing curbside takeout by offering one family-style meal option. A few weeks later, they might expand to selling à la carte side dishes. A few weeks after that, they could expand to a full menu. Along the way, they refine their customer pickup logistics and kitchen processes.
- **Test sites.** If you're testing the use of UV lamps to disinfect hotel key cards, you could try the process in just one hotel before committing to an organization-wide shift — or at just one reception desk station while others use disinfectant.

If your team is accustomed to sharing unfinished work for scrutiny on a regular basis, these approaches will probably be easier for you. If not, you'll need to do more to foster a team culture of risk-taking and learning on your team. For more, see [6 ways to give your team space to speak up, take risks — and ultimately perform better](#).

6. Measure your tests with direct observation (if possible) and by asking good questions.

How do you judge if your hypothesis is valid? To be as objective as possible, try to use direct observation of behavior or results (e.g., did someone buy it?) rather than a hypothetical (e.g., a survey question asking “Would you buy it?”), which may not be a reliable indicator of whether someone would actually use or buy something. You may already have information streams in place that you can tap to gauge the impact of your experiment.

These streams could include:

- Receipts and/or sales trends
- Help desk tickets and/or complaint logs
- Website and email analytics
- Social media engagement
- Recorded calls
- Customer praise and criticism passed along by front-line workers to use in conjunction with other data (don't rely only on anecdotes)

If your team plans to supplement existing data by posing questions to beta testers, put careful thought into the questions to be sure that they yield answers that will help you improve your plan. For example, instead of asking your testers questions likely to yield generic or yes-or-no answers (e.g., “Do you like this?”), try questions like:

- What did it help you do?
- I noticed you paused when you looked at X. What were you thinking then?
- How does this compare to the last time you did Y?

And follow up with the magic question “*Why?*” in order to dig deeper.

7. Embrace failure. Change your mind cheerfully.

Effective experimentation demands humility. As a leader, you need to constantly remind yourself and your team that your desire to *do right* for users is way more important than your desire to *be right*. If you don’t, you risk letting your egos derail your decision-making and, ultimately, the team’s results.

Here are a few ways to cultivate this flexible, innovation-friendly mindset:

- **Express excitement when experiment results debunk your opinions.** Contrary evidence gives you the opportunity to course-correct before it’s too late. That’s gold. A comment from you like *“This is awesome — we’ve learned something fascinating about our users that will help us going forward”* signals that experimentation is for real on your team, not “innovation theater” in order to say you’re data-driven when you’re really not.
- **Reward team members for being data-driven even (or especially) when the data points to failure.** For example, when a direct report’s test email subject line reduces engagement: *“I’m so glad you tried that subject line — we’ve learned something about our users’ top concerns. Thank you for being data-driven.”*

Teams that get punished for misses learn the unfortunate lesson that innovation isn’t worth the risk and they should stop trying.

8. Evaluate your innovation process so you can improve next time.

This step may seem like it would take time that your busy team doesn't have. But consider: A team who experiments carefully and efficiently gets reliable results faster. Think of scrutinizing your processes as an investment in your team's future ability to innovate and excel.

For example, maybe your team realizes that the client interview data that took precious time to collect doesn't yield much because, in the rush to start interviewing, the team failed to include questions about some critical areas. Or your team realizes only after lengthy experimentation that a quiet team member was right in their early critique — and that you need a system to be sure that dissenting views get full consideration from the start.

To unearth these kinds of insights, frequently ask your team in both group and 1-on-1 settings, “[*What's one thing we could be doing better?*](#)” and keep notes on what you hear. And once you've completed an experiment, schedule time for a [debrief meeting](#) so that people can compare their observations on what worked, what didn't, and what to do differently next time. Then, act on what you learn.