

How to integrate continuous user testing into your product development workflow

An interview with Edmond Lau, author of the book
The Effective Engineer and software engineer at Quip



Introduction

Here's an interview with Edmond Lau, author of the book [The Effective Engineer](#) and software engineer at [Quip](#)—a modern productivity suite centered around team communication—to talk about how Quip integrates continuous user testing into their product development workflow. Edmond will show you how to:

1. Iterate faster and more incrementally toward the right product to ship
2. Quickly validate that you're moving in the right direction and build confidence in your product decisions
3. Use persuasive data from real users to help resolve heated design discussions over the hypothetical benefits of various approaches

Q & A

1. Many people think the practice of usability testing is just for UX designers and researchers. Why is user testing important for product and engineering teams?

User testing is a tool, just like automated tests or continuous deployment. And like any other tool that your team invests in, it takes time. Many product and engineering teams ignore user testing because the time spent conducting and learning from user tests is time not spent writing software.

But when we consider all the time we spend tweaking and perfecting features that never get the adoption we want, it's clear that we could be building products much more effectively. The earlier we can make course corrections, the less effort we'll waste and the bigger the impact we'll create. User testing, and in particular *continuous user testing*, is a powerful technique that can help teams more rapidly and incrementally iterate toward the right product to ship.



2. What is “continuous user testing”?

For several weeks during the most recent product redesign at Quip, we ran tests with UserTesting. My team and I would huddle on the couches, to watch the latest batch of videos of users testing out our product. **Watching these user videos became a core part of our workflow and quickly evolved into one of our highest-leverage activities.**

Our product team developed a healthy cadence where we would iterate on a part of the user experience based on some hypothesis, deploy the changes to production, and then run a user test on the new version. Within the hour, we’d receive detailed feedback from real people and watch 30-minute long videos of people making sense of—or failing to make sense of—our changes. Based on our learnings, we’d brainstorm solutions to eliminate sources of friction and confusion and repeat the cycle. **At our peak, we ran up to 12 user tests in a single week.**

We call this technique—where we would aggressively and iteratively run user tests to validate hypotheses, as a normal part of our workflow—*continuous user testing*. It played a key role in guiding our redesign, and it’s a technique that could benefit other teams as well.





3. What is the advantage of “continuous user testing” over doing a larger research project a couple times a year or only around major releases?

Suppose you’re building software, but you could only compile or test your code a couple times a year. You’d have a significant handicap because you’d have to keep building on top of functionality that you weren’t sure even worked. And you’d probably throw away most of what you built.

You give yourself a very similar handicap if you only conduct user testing a few times a year—particularly if the product you’re building isn’t one that you use yourself. If you keep extending or tweaking a product based on your own hypotheses, but without user data, how confident can you be that your assumptions match reality? And **if you run user tests only as an afterthought, and discover that your initial hypotheses are invalid by the time you’re about to ship a major release, the cost of changing course is enormous.**

Continuous user testing helps you build confidence in product decisions at low cost. It lets you debug user behavior just like you’d debug your code’s behavior. For our latest redesign at Quip, we’d keep asking ourselves, “What’s the next piece of functionality where we would learn the most by testing on users?” And then we’d test those areas. The tests confirmed promising design choices and surfaced areas—sometimes surprising ones—that people found confusing. Those validated learnings helped us to confidently ship a redesign, and the launch largely received positive feedback.





4. Can you give a brief explanation of your history with usability testing. How has it evolved and how does UserTesting differ from other forms of usability testing you have done in the past?

User testing is by no means a new concept.

When I worked on search quality at Google, our team would periodically run tests in the usability lab. In a typical test, a researcher would guide a paid volunteer through a list of tasks, and the team could either observe through a two-way mirror or watch the recorded video afterwards. Eye-tracking lasers followed a participant's gaze as it danced across pages, and overlaying this optical data later over the recorded screen provided valuable insights into where people focused their attention.

Once, instead of running a lab test, we even drove out to volunteers' homes to observe how people used Google in their natural habitat.

These user tests often surprised us. Who would have known that some people copied and pasted search results into Word documents and printed them out when doing online research?

That's not an insight any amount of team discussion would have revealed. But while useful, **the overhead involved in scheduling and running an in-person test often meant that user testing rarely became an integrated part of a team's development workflow.** Many teams would conduct tests only as an afterthought, after they had already invested substantial amounts of engineering and design work.

Outside of Google, the teams that actually run user tests continuously as part of their workflows tend to be ones that are scrappy and have low numbers of users—traits that you'd find in early startup teams.

For example, when Akshay Kothari and Ankit Gupta were initially building the Pulse News reader app (acquired by LinkedIn in 2013 for \$90M), they would set up camp in a Palo Alto café and invite visitors to test out new prototypes on an iPad.

Based on the usability issues that people hit, they would then make hundreds of small iterations per day—from fixing interaction patterns to adjusting button sizes. Within a couple weeks, people went from calling it "crap" to asking if it was preloaded on the iPad.

The advent of virtual usability labs and the speed with which you can get results change this dynamic, making these types of workflows accessible to more teams.



5. What are some of the internal benefits of user testing when prioritizing features and making product decisions?

The most valuable benefit is a faster iteration cycle. Teams can iterate faster and more incrementally than with a more traditional build-and-launch approach or oftentimes even A/B tests.

Continuous user testing also affords two other key benefits. First, **it provides a powerful channel to quickly validate and build confidence that you're moving in the right direction.** Driving a product forward based on vision and intuition is important, but we're oftentimes so immersed in our own mental models of how something works that we can become fairly disconnected from someone less familiar to the product. Gathering feedback from real users helps ground our reality.

Second, **the persuasiveness of data from real users helps resolve discussions.** How often have we spent hours in heated design discussions over the hypothetical benefits of various approaches, only to move forward based on some line of argument that doesn't necessarily have everyone's buy-in? We often spend our energy debating because we don't have enough data, and continuous user testing gives us another tool to collect that data. And that means we can focus on results and ship what actually works rather than what hypothetically works.





6. What is the relationship between user testing and more quantifiable experimental methodologies like A/B testing?

A good starting point for continuous user testing is where A/B testing falls short. Sometimes the interactions we're trying to understand are too complex to distill into a single number. Or perhaps measuring it would take a significant amount of time and traffic to detect any meaningful difference. How much of the core product do new users understand after going through a short tour? How obvious are the interactions required to accomplish a common task?

It's tough to get deep answers to these questions with an A/B test.

Continuous user testing can help in these situations. By observing where users succeed and stumble as they work through a series of tasks, we can gain a deeper understanding of user behavior that isn't captured through high-level metrics. **That understanding can validate your intuition and your hypotheses, as well as surface unexpected sources of friction.**

Interactions where users don't require a significant amount of context to understand what's going on generally work better for rapid user tests. For more complex, power-user features for Quip, we've relied more heavily on internal dogfooding or beta tests with customers.

All told, the many hours that we spent watching users understand and work with Quip introduced a level of confidence in our redesign that we wouldn't have otherwise. Many of the ideas we tried didn't pan out, but continuous user testing let us focus and ship the ones that did.

About UserTesting

UserTesting enables every organization to deliver the best customer experience powered by human insight. With UserTesting's on-demand Human Insight Platform, companies across industries make accurate customer-first decisions at every level, at the speed business demands. With UserTesting, product teams, marketers, digital and customer experience executives confidently and quickly create the right experiences for all target audiences, increasing brand loyalty and revenue. UserTesting has over 1,200 subscription customers including 48 of the top 100 brands in the world, and has delivered human insights to over 35,000 companies to-date. Backed by Accel and OpenView, UserTesting is headquartered in San Francisco, CA. To learn more, visit www.usertesting.com.



HEADQUARTERS

690 5th Street
San Francisco, CA 94107
USA

—

WEB

UserTesting.com

CONTACT US

1-888-877-1882