LEARN@LIFT:

Episode 3

Rapid Prototyping & User Testing To Validate Your Ideas

COMCAST NBCUNIVERSAL | LIFT Labs

In partnership with GAN

CUSTOMER DISCOVERY: An Iterative Process – Episode 3

TITLE:

Rapid Prototyping & User Testing To Validate Your Ideas

DESCRIPTION:

Learn how to conduct simple tests that can help confirm your solution is the right one, before spending the time and money to actually build it.

[LEARN@LIFT Episode Intro]

In this episode, we're going to discuss turning your low-fidelity solution sketch into a functional prototype, plus a way to test your product's hypothesis, before starting the development process.

Now, I just mentioned a hypothesis. This is key. By moving from rough sketches to the process of forming a final solution design, you've made an implicit assumption. That you've sketched the right solution that's worth investing resources in.

But to be certain, it's best to turn that assumption into an explicit statement that can be tested to answer two different questions:

1. Did we pick the right solution?
2. Are we building the solution in a way that users will like?

To answer the first question, we'll use a Minimum Viable Test — not a Minimum Viable Product — to confirm that the positive feedback users gave us on our sketch, or low-fidelity prototype, was in fact on the right track.

To answer the second question, we'll move to building out a High-Fidelity or Hi-Fi prototype that solidifies how users want to interact with the product.

So, let's start with the idea of a Minimum Viable Test.

If we're to continue our example from last episode, our hypothesis may follow an If-Then statement that looks like this:

> If we generate ready-to-write SEO Blog briefs for marketing teams, then they'll be able to generate more organic leads.

We may also have a series of hypotheses that we've created along the way. For example, a second hypothesis may be:

> If we generate ready-to-write SEO Blog briefs for marketing teams, then they'll be more likely to dedicate the staff time to writing new blog posts.

If we want to test both of these hypotheses, we'll need to design an experiment that measures two things:

1. Did our users actually invest the time to write new blog posts when given an SEO brief?
2. Did that blog post help drive new leads? Did we give them valuable keywords to use?

You'll notice that what we're starting to test for is how effective we are at changing or working alongside user behavior, and if that behavior successfully accomplishes the job they're trying to get done — while solving the pains they're willing to pay to alleviate. So, we're going far beyond just words and user feedback. We're looking at solid evidence based on behavior.

The most common type of test is the Wizard of Oz test.

Its name comes from the book and movie, where the great and mystical Wizard of Oz turns out to be nothing more than an ordinary human being hiding behind a curtain concealing his true identity. The "wizard" — or in this case, you — is actually making the product work behind the scenes, often by hacking together third-party tools that don't require any coding.

For example, to get started with your 'test product' a B2B marketing team may simply create a guest login to their customer survey tool, where you'll be able to download a list of survey responses each week.

Then, you might pull that content into a spreadsheet, find repeated words using an Excel formula, and then run those words through an SEO database like Ahrefs, before manually writing up a series of H1, H2, and H3 sentences to form a new blog post outline.

Then, if the marketing team you served up those SEO briefs to end up writing and posting new content to their blog, you're in luck. Better yet, if they took that action inside a set timeframe you were expecting, you're onto something.

Note, this type of testing is particularly useful if you're building anything that's based on AI, data analytics, or any other complex process that may require a significant amount of time or capital to build.

As you design your test, make sure to stick to designing the test around the single, greatest assumption you're making in the design of your solution. You will almost always get 2-3 risks tested inside one run, but you should have a primary goal for your test to ensure you leave the test with conviction and direction, pushing you one way or another.

After each test you run, ask yourself: what other risks should I be testing for? You can always layer on multiple tests, one after another in order of riskiness, to increase your conviction level.

Next, let's continue with the idea of a Hi-Fi prototype.

You're feeling confident you're on the right track, so now, it's time to get a production-ready first version of your product.

This is where, if you don't have a design or product background, you'll want to bring in a professional UX or product designer.

If you don't have a contact in your network with this type of background, sites like Toptal, Upwork, and product agencies are going to be your next set of conversations. In the next episode, we'll include some thoughts on finding the right product development partner for your startup.

The major advantage of a Hi-Fi prototypes is more realistic interactions with your users. For example, when a user is clicking around to find the right screen in a software design, or grabbing a physical product, the design actually responds to how a user is engaging the product.

The goal of building in Hi-Fi interactivity is testing your user's workflow, specific visual components, or the product layout and hierarchy. Often, these prototypes often look just like the real thing to users, which means your test participants will also behave as they actually would as your customer.

Just like in your earlier interviews, and the "build-learn-adapt" loop we discussed, you'll want to go back to your users — and ideally, an even wider group of people – to watch them interact with the product. Inside these interviews, begin with the basics of what you're hoping the product will allow them to achieve, and how the product is laid out.

From there, allow them to engage with the product on their own terms, as if you weren't around. Then, note where they hit questions and need your guidance to instruct them on the next steps.

Clarify your goals, and ask if they have any suggestions for how to layout that specific element of the product more effectively. After a handful of interviews, you'll start to observe patterns, and have a handful of alternative designs ready for them to consider.

Once you've run this build-learn-adapt loop a few times through — usually three times — you'll be ready to build the first version of your product to deliver to the users who have been building your solution alongside you. We'll cover what to do to get started in our next episode.

[Insert LIFTLabs Outro]