TESTING PRODUCT IDEAS HANDBOOK

ITAMAR GILAD

itamargilad.com

TOC

Introduction

What Do We Need To Validate?

The AFTER framework

Assessment

- Goals Alignment
- ICE analysis
- Business Modeling
- Assumption Mapping
- Stakeholder Reviews

Fact Finding

- Data Analysis
- User/Customer Interviews
- Field Research
- Surveys
- Competitor Analysis

Tests

- Minimum Viable Product
- Smoke Tests
- Usability Tests
- Human Operated Tests
- Labs
- Beta Programs and Previews

Experiments (Split Tests)

❖ A/B Tests

❖ A/B/n and Multivariate Tests

Release

❖ Release Tests

Interpreting Results

The Need For a System

About Itamar Gilad

Introduction

"If you want to have good ideas you must have many ideas. Most of them will be wrong, and what you have to learn is which ones to throw away" -- Linus Pauling

There's never a shortage of product ideas. Managers, stakeholders, team members, customers, even friends and family will tell you what you should do with the product, and they all expect you to take notice because they think their idea is good. But there's a reality that is seldom considered:

- The vast majority of product ideas lead to no measurable improvements in business results or value-to-customer. Some actually cause negative results. [1]
- No one, no matter how senior or experienced, can predict which ideas will work and which won't - there are just too many unknowns.
- Most companies use weak heuristics, opinions, and archaic decision processes to place bets on a handful of unproven ideas. This is arguably the biggest source of waste in the industry.

Therefore It is our job to investigate as many product ideas as possible. For each idea we need determine that it is valid and worth investing in as quickly as possible. This a core principle of Lean Startup, Design Thinking, and other modern product development methods, and the default mode of operation for many successful tech companies today. Still it's a concept that is widely misunderstood and misused.

In this ebook we'll go over what product idea validation means and how we go about doing it.

What Do We Need To Validate?

Ideas can fail in a variety of ways. Renowned product management executive and coach, Marty Cagan, argues we need to look at four areas of risk:

1. **Value**

   The product idea addresses real needs in the market; there is a demand for it.

2. **Usability**

   Users will be able to start using the product quickly and without much friction

3. **Feasibility**

   We're able to build the product idea within reasonable time and cost.

4. **Viability**

   The idea makes business-sense and thus not create risks for current business.

Naturally we don't have to validate every idea against all four risks. Feasibility and business-viability will not be a problem for most product features. Usability issues are very often fixable. So while we need to watch out for the other risks, we need to pay closest attention to the low-value risk.

The AFTER Framework

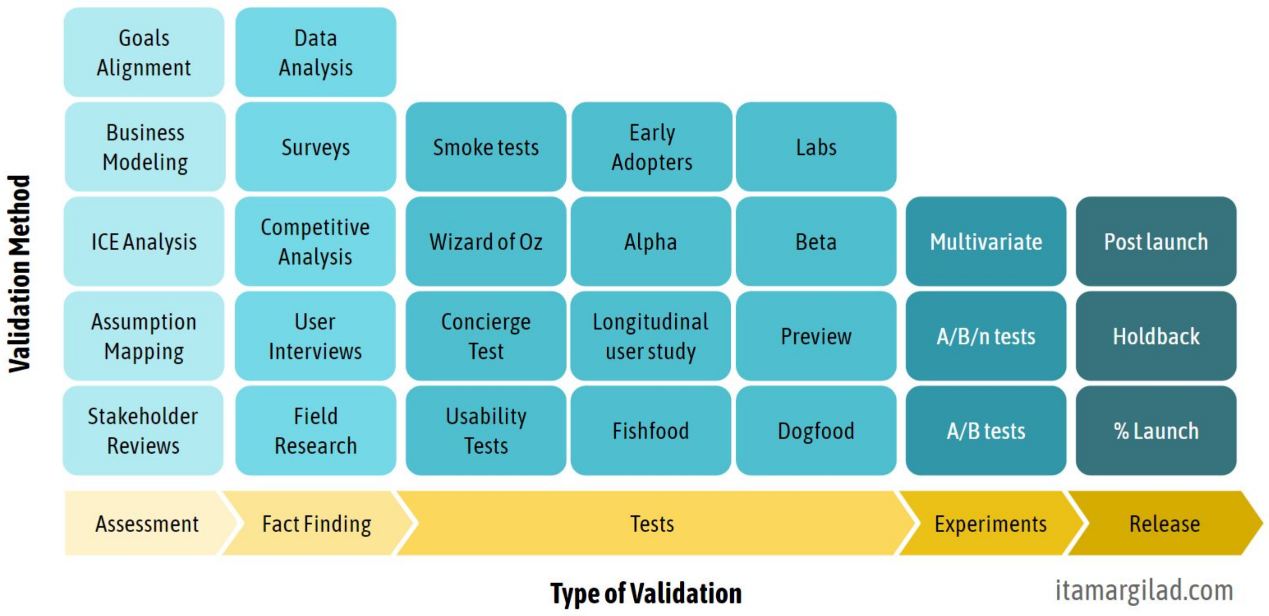
“If it disagrees with experiment, it's wrong.
In that simple statement is the key to science.”
— Richard Feynman.

Inspired by the scientific method, we want to ensure that our ideas are valuable, usable, feasible and business-viable by conducting *research* and collecting *evidence*. Our confidence in our ideas should stem from supporting evidence, rather than from opinions.

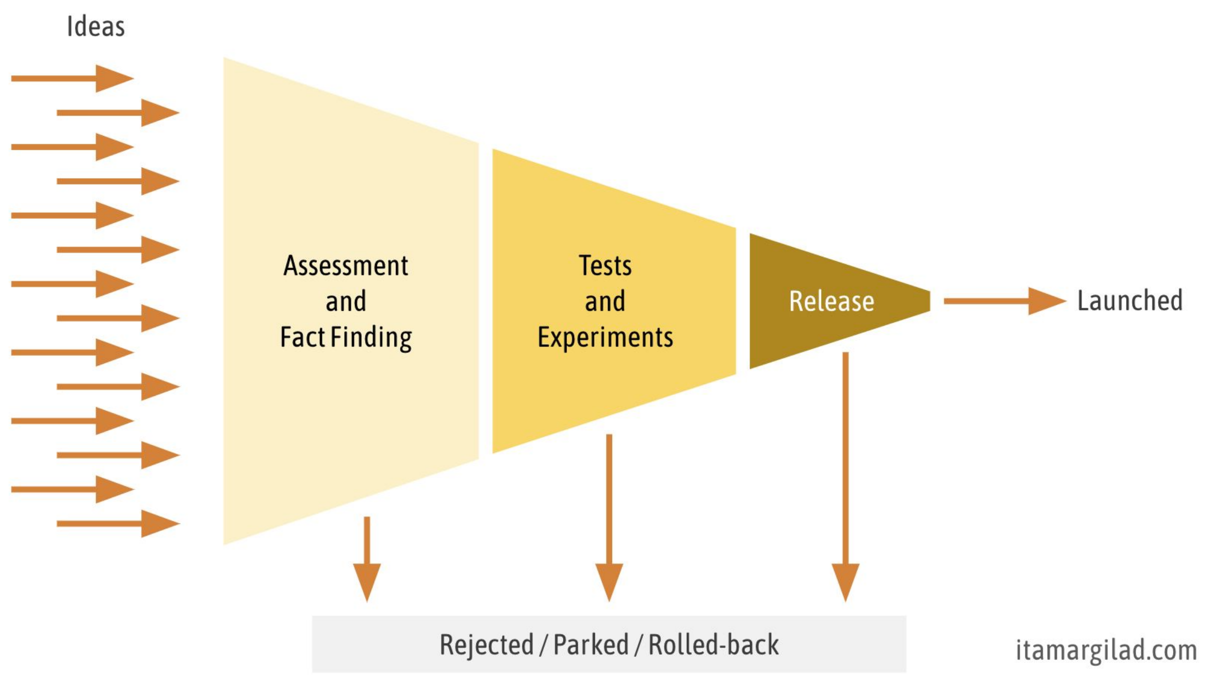
Our mission is therefore to start collecting evidence and using it to determine which ideas are showing most promise and deserving of more investment. Broadly we need to conduct five types of research activities captured by the acronym AFTER:

- **Assessment** - Quick and rough evaluation of the potential and risk of an idea with no external research
- **Fact Finding** - Actively looking for data and facts that may support or refute the validity of the idea
- **Tests** - Putting increasingly more evolved versions of the idea in front of users and measuring their reactions.
- **Experiments** - Conducting quantitative tests that control against false results.
- **Release results** - Gradually releasing the product to more users and customers while monitoring the reaction.

The diagram below shows some of the most important validation methods in each category, but there are many others, and more are being invented as we speak. Make no mistake - every product idea can be validated. Idea validation is really only limited by our creativity and willingness to step outside our comfort zone.



We will not use all five types of research with every idea - that will be prohibitively time consuming. Instead consider your research activities as composed of three funnels. We want to use assessment, and often also fact-finding quickly across many ideas - the more the better. Most ideas will be rejected or parked (ie deferred to be looked at again at a later time). The minority that pass will go on into testing and experimentation. Many more ideas will be rejected or parked at this phase, but hopefully some will be good enough to go into the next phase - Release. In the release phase we complete the development of the product change so it is production-ready, then we may start rolling it out gradually and observe the results. We consider the launch itself a type of test, and in some cases may even choose to rollback a partially/fully launched idea due to negative results (typically for fixing and re-launch).



Flow of ideas through validation funnels

Cheap, low-risk ideas can progress through the funnels very quickly, while costly, high-risk ideas will require a lot more validation. That doesn't necessarily mean that they will take longer to launch compared to the "just ship it" approach - as we validate ideas we also develop them, and often in the process we learn that we can reduce the scope of the idea and release it faster.

Next, let's take a close look at each research category and its validation methods.

Assessment

Idea assessment is all about determining quickly, with no external research, if an idea is worth moving forward with at this time. Assessment is never enough to choose to build and launch an idea (although that's the practice in many companies) as it only gives us weak evidence. Still the techniques below help us evaluate the idea more objectively and in a structured and somewhat detached way, and so can act as important filters.

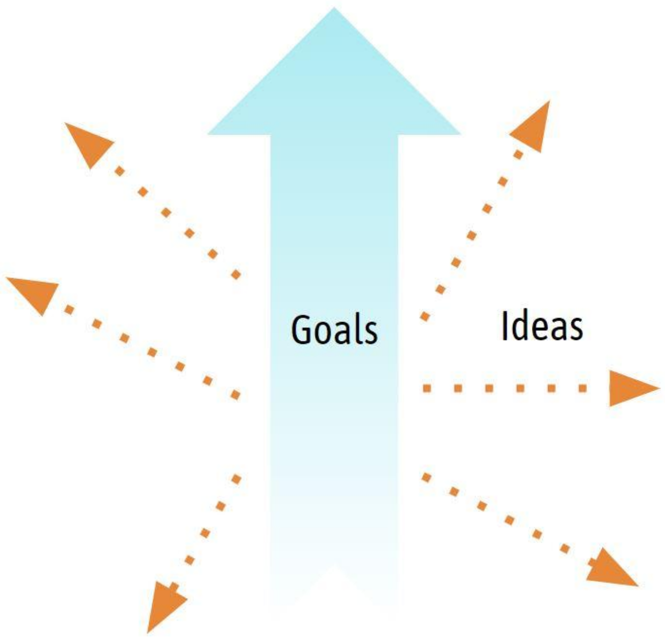
Photo by Mikechie Esparagoza from Pexels

Goals Alignment

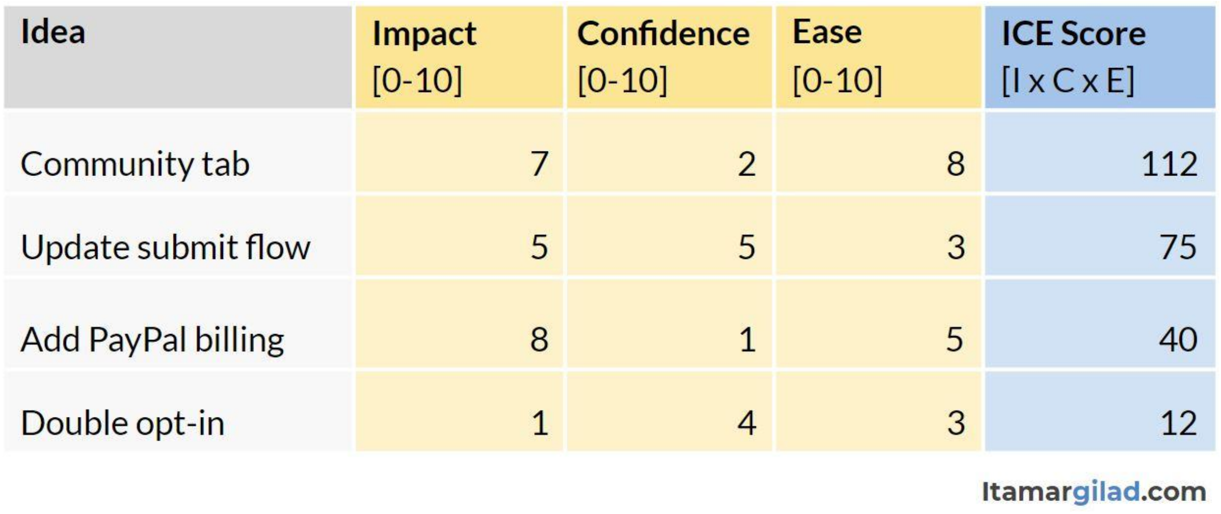
Is this idea helping us achieve any of our goals? If not does it represent a new opportunity that should be covered by the goals? If the answer to both questions is no, we're probably better off parking the idea and maintaining focus.

Imagine for example that your company decided to focus on growing its share in the small-medium business market segment. Some time later an idea lands on your desk - build a new support tool to streamline the work of customer support. The idea has the potential to save time and cost, however SMB clients very rarely buy packages that include customer support, therefore this idea has lower potential to impact the company goal.

Having clear, focused, measurable goals is important, of course. I recommend using hierarchical metrics trees and OKRs to ensure goals are set, communicated and aligned across the company.



ICE analysis



Idea	Impact [0-10]	Confidence [0-10]	Ease [0-10]	ICE Score [I x C x E]
Community tab	7	2	8	112
Update submit flow	5	5	3	75
Add PayPal billing	8	1	5	40
Double opt-in	1	4	3	12

Itamargilad.com

ICE, invented by growth expert, Sean Ellis, is a technique to score ideas based on three criteria:

- **Impact** - how much value will the idea deliver to the market (and consequently to the company)
- **Confidence** - how much evidence do we have that this idea will have the expected impact.
- **Ease** - Essentially the opposite of effort, for example an idea that takes 1 week to build is considered “easy” and therefore gets an Ease rating of 10, while an idea that will take 6 months to build is just a 3. The scale should be set by the company or team.

ICE scores themselves are meaningless. The goals of using ICE are:

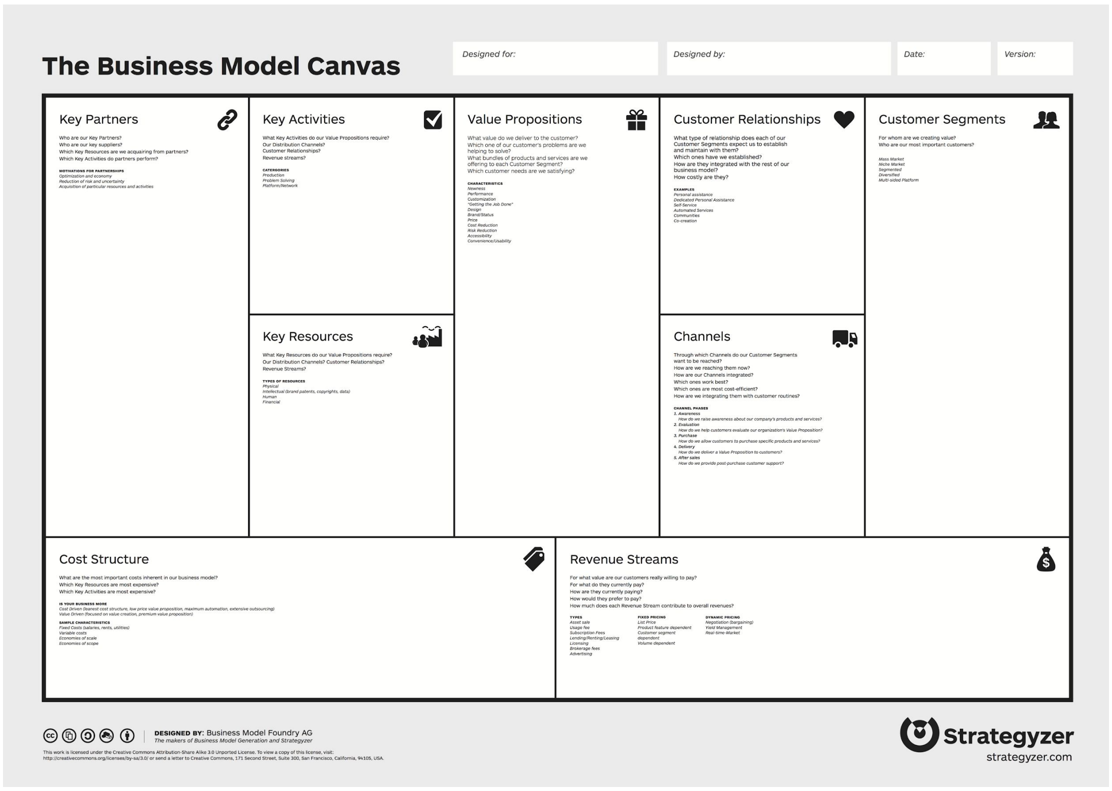
- Asses the idea in a more structured and unbiased way
- Compare this ideas to others
- Keep a running score as we learn more about the idea

Business Modeling

When evaluating an idea for new products or new business models, it's good to first check that the idea is business-viable on paper:

- Will the idea generate meaningful profit? (note: not every idea is supposed to)
- Do we have the necessary partnerships and resources in places to make this idea happen?
- Do we have the right channels to reach potential customers?

We can validate business viability using a simple revenue/cost projection in a spreadsheet, Strategyzer's business model canvas, Ash Maurya's Lean Business Model Canvas, or other business analysis techniques.



The Business Model Canvas

Designed for:
Designed by:
Date:
Version:

Key Partners

Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquairing from partners?
Which Key Activities do partners perform?

MOTIVATIONS FOR PARTNERSHIPS
Optimization and economy
Reduction of risk and uncertainty
Acquisition of particular resources and activities

Key Activities

What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

CATERGORIES
Production
Problem Solving
Platform/Network

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

CHARACTERISTICS
Newness
Performance
Customization
"Getting the Job Done"
Design
Brand/Status
Price
Cost Reduction
Risk Reduction
Accessibility
Convenience/Usability

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES
1. Awareness
How do we raise awareness about our company's products and services?
2. Evaluation
How do we help customers evaluate our organization's Value Proposition?
3. Purchase
How do we allow customers to purchase specific products and services?
4. Delivery
How do we deliver a Value Proposition to customers?
5. After sales
How do we provide post-purchase customer support?

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platform

Cost Structure

What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

IS YOUR BUSINESS MORE
Cost Driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing)
Value Driven (focused on value creation, premium value proposition)

SAMPLE CHARACTERISTICS
Fixed Costs (salaries, rents, utilities)
Variable costs
Economies of scale
Economies of scope

Revenue Streams

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

TYPES	FIXED PRICING	DYNAMIC PRICING
Asset sale	*List Price*	*Negotiation (bargaining)*
Usage fee	*Product feature dependent*	*Yield Management*
Subscription Fees	*Customer segment dependent*	*Real-time-Market*
Lending/Renting/Leasing	*Volume dependent*	
Licensing		
Brokerage fees		
Advertising		

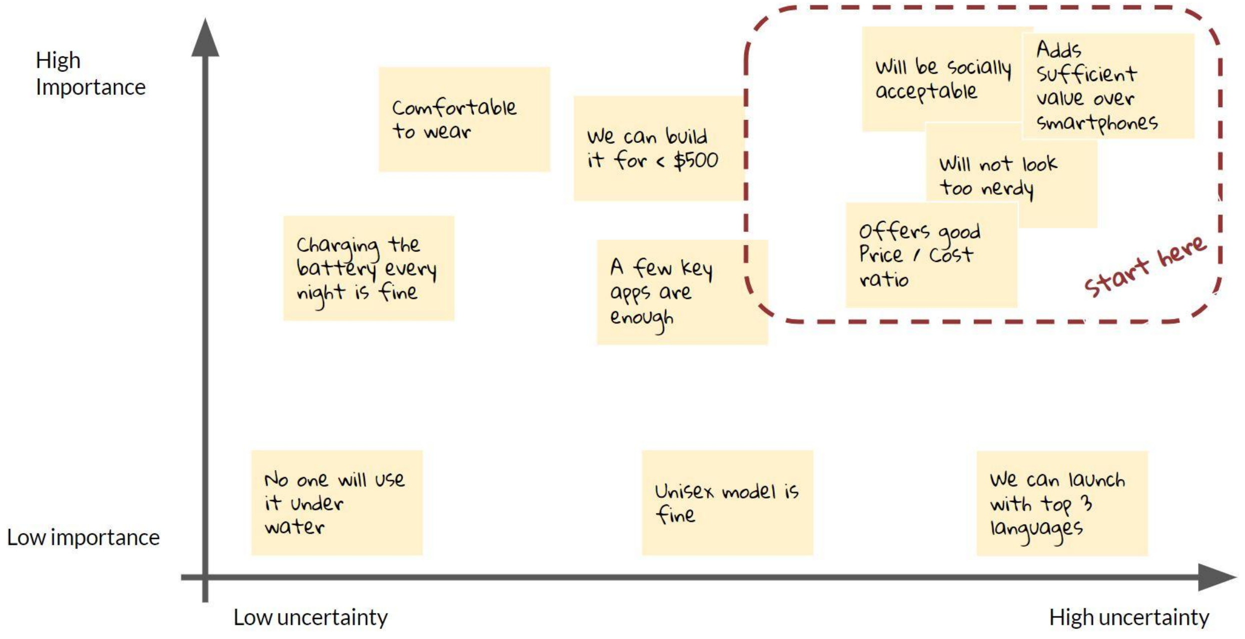
DESIGNED BY: Business Model Foundry AG
The makers of Business Model Generation and Strategyzer

Strategyzer
strategyzer.com

Business Model Canvas. Source: strategyzer.com

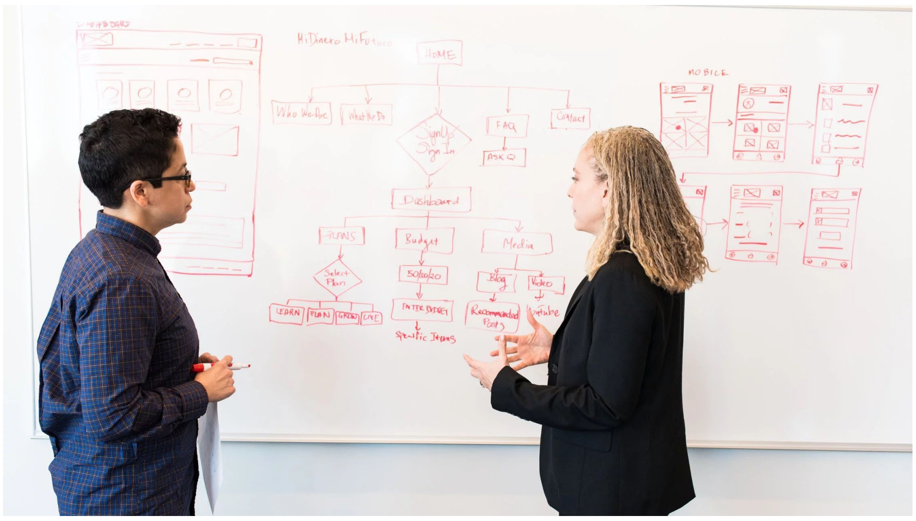
Assumption Mapping

Assumptions mapping, invented by David J. Bland [2], is a team brainstorming technique to surface assumptions and risks in your bigger ideas. Assumption maps place assumptions on two axes - how certain are we about the answer, and how important the answer is. After completing the assumption map you should have a deeper understanding of what might go wrong with your idea, and what you will need to validate first


High Importance
Low importance
Low uncertainty
High uncertainty
Comfortable to wear
We can build it for < $500
Will be socially acceptable
Adds sufficient value over smartphones
Will not look too nerdy
Charging the battery every night is fine
A few key apps are enough
Offers good Price / Cost ratio
Start here
No one will use it under water
Unisex model is fine
We can launch with top 3 languages

Assumption map

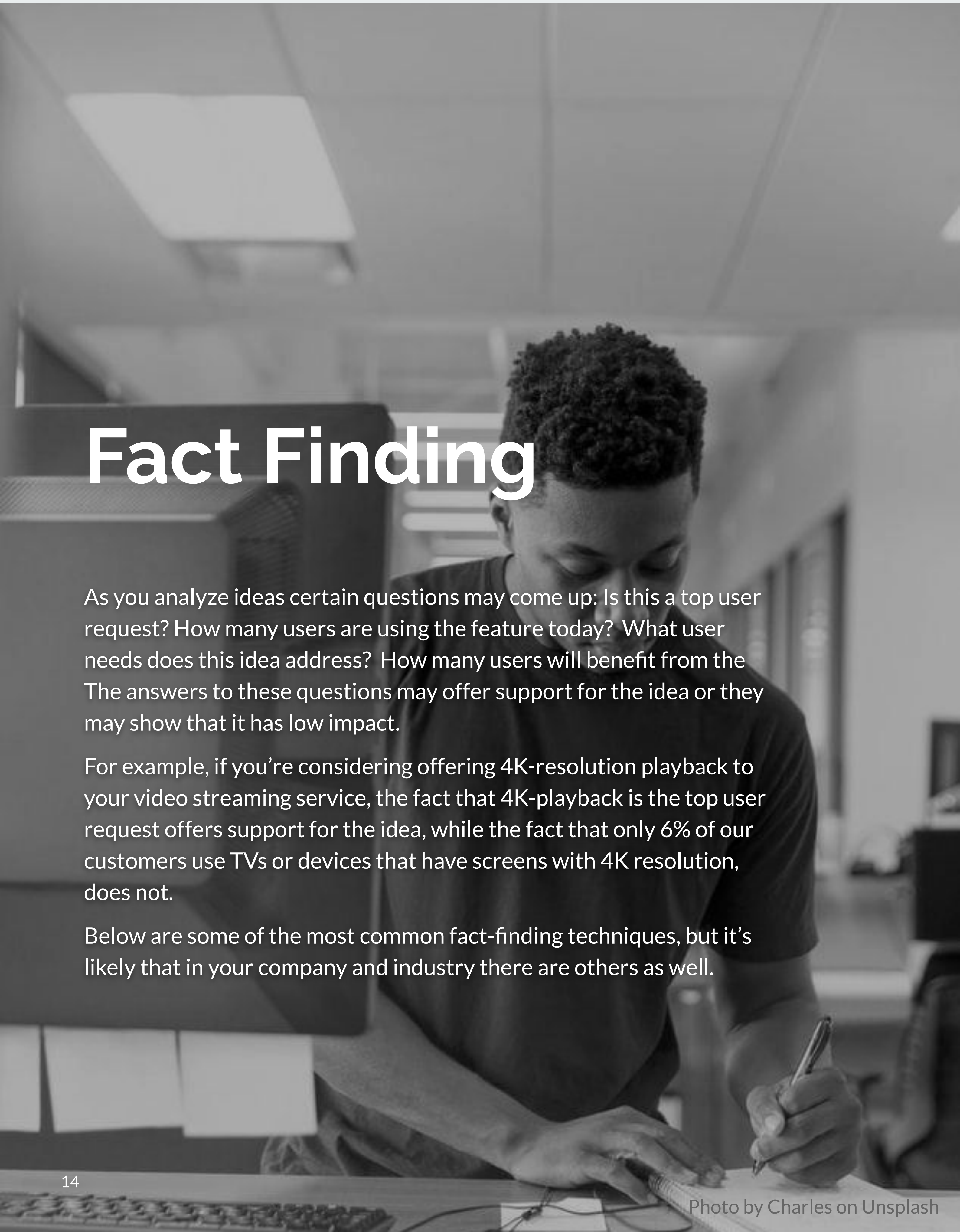
Stakeholder/Partner Reviews



Many ideas, even potentially very good ones, die because they run into internal opposition from stakeholders. Sometimes the idea is perceived to be competing with an existing product or initiative. Other times it might might seem to create a risk: legal, PR, brand, security, privacy, that that the owner of the area will deem unacceptable. Or it may be that the idea depends on the stakeholder to take action.

It's therefore a good idea to review your idea with relevant internal stakeholders early, so not to be surprised at a late stage, and not to surprise them. It's best to hold these reviews one-on-one, to avoid groupthink and forming of oppositions. The objective is not for them to validate the idea has value (they will definitely try to tell you), but to flush out risks and to discuss ways to test if the risk is real and if so how to mitigate

A variant of stakeholder reviews is Partner interviews, where you run the idea by a partner to check mostly the same things.



Fact Finding

As you analyze ideas certain questions may come up: Is this a top user request? How many users are using the feature today? What user needs does this idea address? How many users will benefit from the The answers to these questions may offer support for the idea or they may show that it has low impact.

For example, if you're considering offering 4K-resolution playback to your video streaming service, the fact that 4K-playback is the top user request offers support for the idea, while the fact that only 6% of our customers use TVs or devices that have screens with 4K resolution, does not.

Below are some of the most common fact-finding techniques, but it's likely that in your company and industry there are others as well.

Photo by Charles on Unsplash

Data Analysis

They say that actions speak louder than words, and the same is true for your users and customers. Often the first step is to analyze data that shows the behaviors and attitudes of your users.

Data sources include:

- Log data
- Clickstreams
- Funnel analytics
- Screen heatmaps
- Session replays
- User feedback
- Customer requests
- Profile data
- CRM data
- Anything else coming from users/customers

With data we answer questions like:

- What % of users use the service with 4K-resolution screens
- How many users try to recover their password per week?
- How many customers have asked for 2-factor authentication? How many seats does this represent.

Finding a few data points in support of the idea, for example, the last two promos were a success, forms anecdotal evidence. Our brains can easily over extrapolate or find patterns in noise, hence anecdotal evidence is considered weak. Finding consistent patterns in medium-big data sets provides stronger evidence. For example: 10 out of 13 promos we ran in-product over the past 2.5 years yielded a conversion improvement of 3% or more.

Photo by PhotoMIX Ltd. from Pexels

User/Customer Interviews

User interviews is one of the most important tools in the toolset of product teams, and one of the most underutilized. While data analysis can show you how people behave, qualitative research, and especially user interviews can help you understand why they do what they do.

For example, at Gmail we got repeated requests from users in emerging markets to allow them to log-out of the Gmail app on mobile devices. The data was clear, but the reasoning was not.

On mobile "logging-out" of Gmail would mean logging out your Google account across all apps and the operating system itself. This is obviously not what the users wanted, so we repeatedly waved this need away as "the users simply don't get it". Only when we started interviewing users in Africa, South America and South Asia did we learn the reason for the request. Mobile devices are often shared inside the family, and thus leaving Gmail logged-in, would mean that your parents, kids or siblings will be able to read your mail. Android does have a solution for this - User accounts, and we learned that we need to communicate this solution clearer.

Photo by Buro Millennial from Pexels

Interviews can take place anywhere: in your office, in pre/post-sale meetings, in a support call, in a coffee shop, in a hallway or in an industry conference. They can last from 5 minutes to an hour. Every interview is an opportunity to learn something about the background and mindset of current and potential users and customers. You can carry out “problem interviews” - checking whether people and companies have the needs you imagine them to have, and what they do about these needs today, and “solution interviews” - checking what they think of your idea, are they willing to make some commitment to it, or both.

To make the most of interviews you should plan them carefully and have a baseline script. It’s often recommended to have one person conduct the interview while another take notes, and to debrief and summarize the findings immediately after the interview. During the interview it’s best to ask open-ended questions and avoid guiding the interviewee (“Do you agree that tax-filing is very hard?” isn’t a great question). “Show me” is always better than “Tell me” - where possible ask people to demonstrate their answer (“Can I see the last note you created on your phone?”).

Doing a handful of interviews creates anecdotal evidence, which can send you in the wrong direction. The strength of the evidence grows with the number of users interviewed - typically around the 10-12 interviews point you will start seeing patterns emerge. It’s good practice to have a regular set of 5-6 interviews scheduled per week, every week, to test the latest ideas and features in development.

Field Research

Field research observes people in their natural environment and tries to understand their behavior and attitudes in context. Some common types include: Ethnographic Interviewing, Contextual Inquiry, Diary/camera studies, and Follow-me home. Field research is most often carried out by a team, and lead by a user researcher or by a designer.

Photo by Nur Andi Ravsanjani Gusma from Pexels

Surveys

Online surveys are a fast and cheap way to “interview” many users at once, though at a much more superficial level. Surveys can help you learn more about your user base, fill gaps in your data and gauge user attitudes, among other things.

Keeping the surveys short and light on mental load will improve completion rate. Use the language of the user and avoid jargon or loaded terms. Most questions should be closed-ended - multiple choice, checkboxes, grids etc. Typically you’ll want to have no more than 7 options per question and make them as short as possible. Don’t forget to include fallback options : Don’t know, Unsure, Not applicable, None of the Above, Prefer not to say, Other… . Another common survey technique is to offer scaled answers:

Do you currently own a domain (for example johnbeck.com, or myblog.com)?

❏ Yes
❏ No
❏ Unsure

How important is it for you to have your own domain in the web address of your blog - for example johnbeck.com/blog :

❏ Not important
❏ Somewhat important
❏ Very important

You can also include polls in your surveys. The usual caveat that users are bad at predicting their own future behavior, holds.

What of the following features will be most useful for you in your blog (pick one)?

- ❏ A personal domain, for example johnbeck.com
- ❏ An automated grammar checker
- ❏ Help from a professional editor
- ❏ A quick way to do fact-checking
- ❏ None of the above
- ❏ Other: ______________________

In this case it's important that different respondents will see the options listed in different order. (None of the above and Other should stay at the bottom).

You can run surveys in a variety of ways:

- In-product - usually very short surveys only
- Via email
- Via support tools
- Through third party service - Often these services will be able to find respondents for you as well.

However surveys have many pitfalls and can easily generate bad results:

- Sampling bias - Sending the survey to a non-representative sample, or having only certain types of users self-select to answer.
- Misunderstanding - As you're not there to explain the questions and answers, users may misinterpret what you're asking. Simple concise questions, examples and pre-testing are all important.
- Loss of nuance - The respondents are only limited to the selected options. It's therefore a good practice to add some (optional) open questions.

For these reasons we never consider survey results as strong evidence.

Competitor Analysis

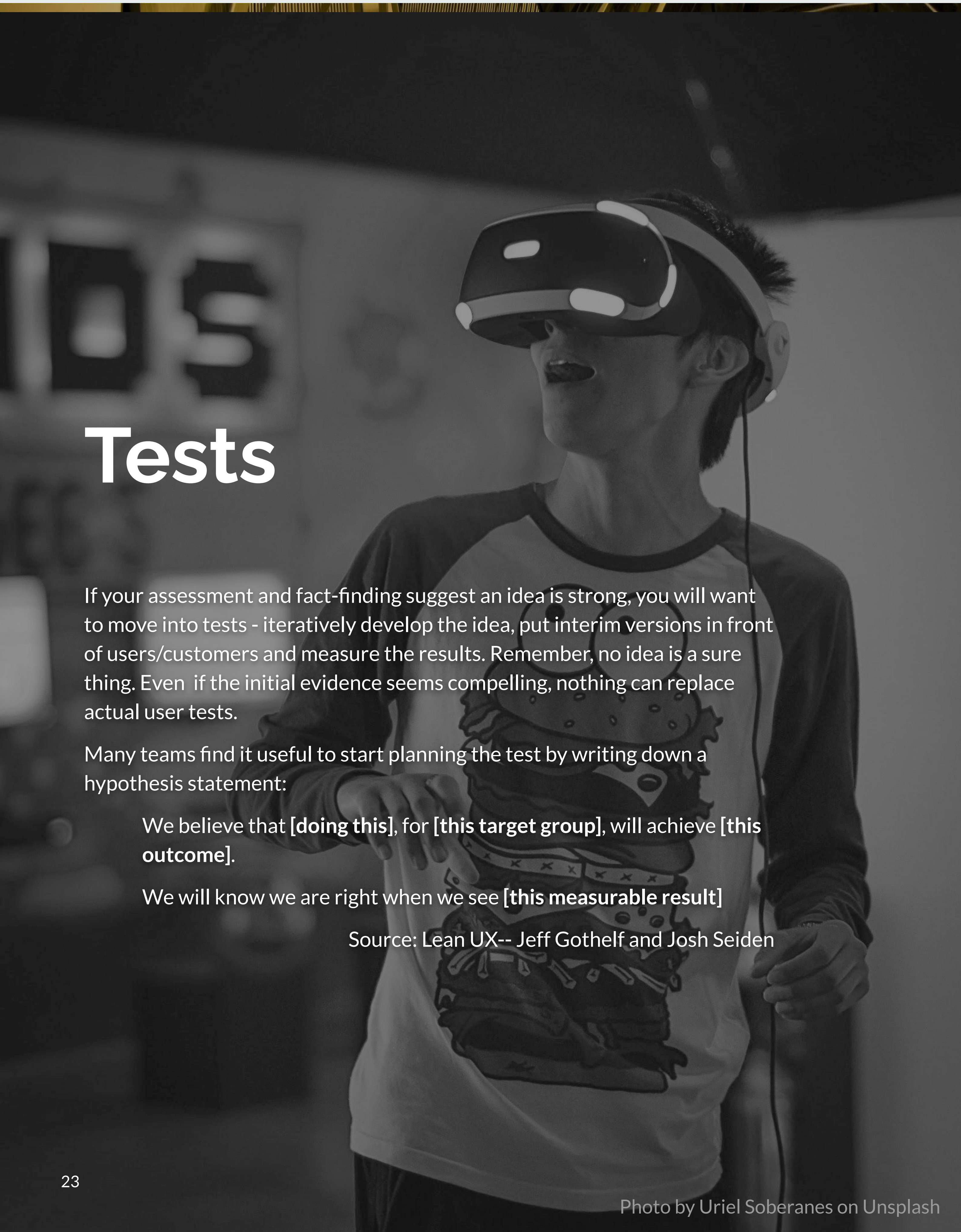
Competitor analysis can be helpful when evaluating ideas, but with clear caveats. Our goal is to learn from our competitors, but not necessarily copy what they have done. It is tempting to assume that an idea is good if a competitor product has it. However, competitors, even very successful ones, don't have a crystal ball or know necessarily something we don't. Obsessing over competitors though can make us a follower, always chasing so-called "must-have" features, yet never creating any differentiation.

When conducting competitor analysis we want to ask the following questions:

- How many competitors attempted to address this problem or opportunity?
- How have they implemented their solution?
- How successful is the solution (adoption, user feedback)? This is harder to find out, but possible from forums, interviews with users of the product, and sometimes the competitor's own messaging ("we're switching off this feature due to low usage.")
- How do they communicate the value to users (what's the value proposition)?
- How do they price it?

If no one attempted to do what we're considering to do, we gain no confidence that the idea is good. If one or two competitors have, we gain low confidence. If all competitors have a similar solution, we see this as a stronger signal from the market, but only gain medium-low confidence — it's possible they all copied from each other or missed the mark.

Sometimes it's useful to broaden our lens and look at *comparable* products as well. Comparables are products that target the same market, but address a different problem altogether. For example if we're developing a travel booking system for enterprise sales reps, and are considering ideas for reporting, we may wish to look at how reporting is done in popular CRM systems.



Tests

If your assessment and fact-finding suggest an idea is strong, you will want to move into tests - iteratively develop the idea, put interim versions in front of users/customers and measure the results. Remember, no idea is a sure thing. Even if the initial evidence seems compelling, nothing can replace actual user tests.

Many teams find it useful to start planning the test by writing down a hypothesis statement:

> We believe that **[doing this]**, for **[this target group]**, will achieve **[this outcome]**.
>
> We will know we are right when we see **[this measurable result]**

Source: Lean UX-- Jeff Gothelf and Josh Seiden

Photo by Uriel Soberanes on Unsplash

Minimum Viable Product (MVP)

The term Minimum Viable Product was first used by aeronautical engineer turned business consultant, Frank Robinson in 2001. Robinson argued that companies should develop a product that is *"big enough to cause adoption, satisfaction and sales, but not so big as to be bloated and risky."*[3] The MVP was designed to create "*maximum ROI divided by risk*." In other words, he defines an MVP is a fully-fledged product that maximizes outcomes yet minimizes output.

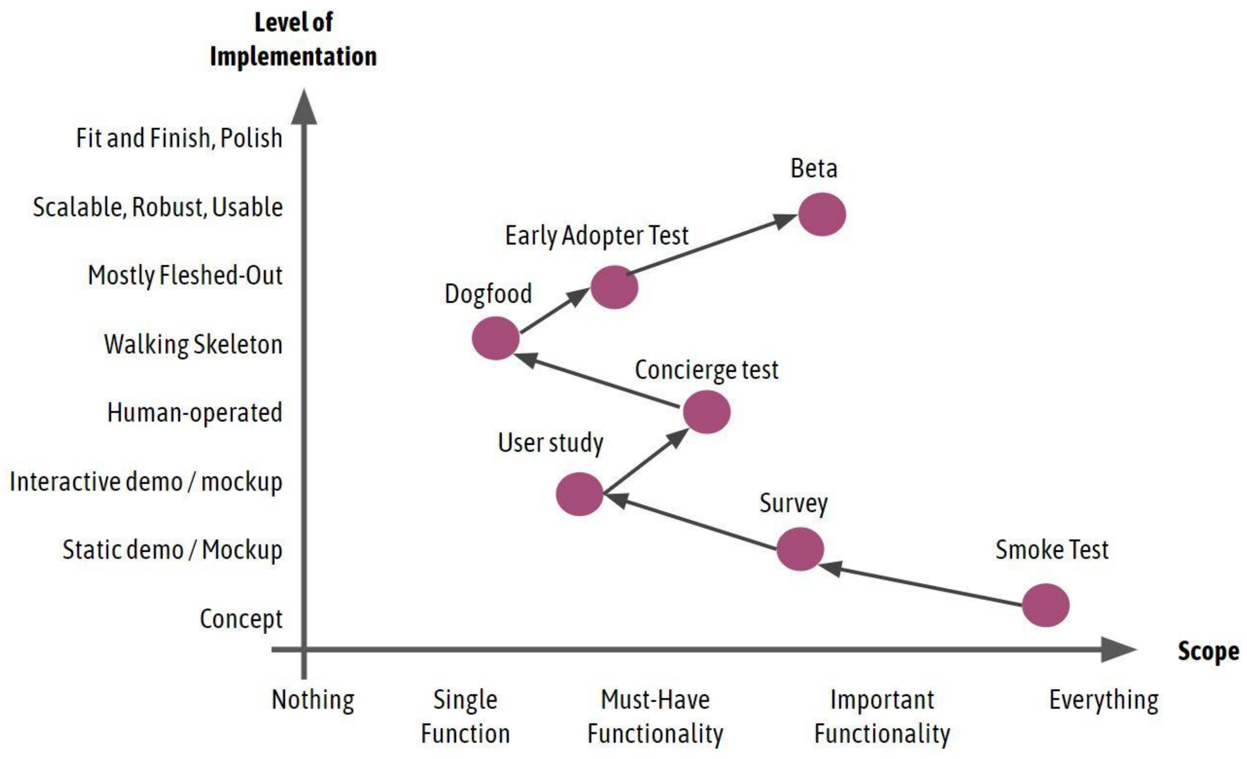
However MVPs entered the mainstream vocabulary only ten years later, with Eric Ries's book *The Lean Startup*. Ries used a different definition:

> *"MVP is that version of the product that enables a full turn of the Build-Measure-Learn loop with a minimum amount of effort and the least amount of development time."*
>
> *-- Eric Ries / The Lean Startup*

Per this definition, many of the validation methods we'll go over qualify as MVPs, and indeed many of them are called out as types of MVPs in the book. In other words, in Lean Startup terminology, Minimum Viable Product is a principle rather than a specific validation technique — always validate with the smallest possible method that produces the evidence you need to learn.

There's often confusion about what constitutes an MVP. Some wrongly assume it's about launching low-quality, incomplete products fast. Others go into full waterfall development of the minimum product we can sell to customers (per Robinson's original definition). Neither of these approaches is correct. At any stage in the development we should decide what's minimal and viable — MVPs in early stages of a product idea should definitely be very light on implementation, while those at the late stages should be much more complete and better written.

There are two main ways we can reduce a product to its minimum - reduce the scope of functionality included and reduce the level or depth of implementation.



Smoke Tests

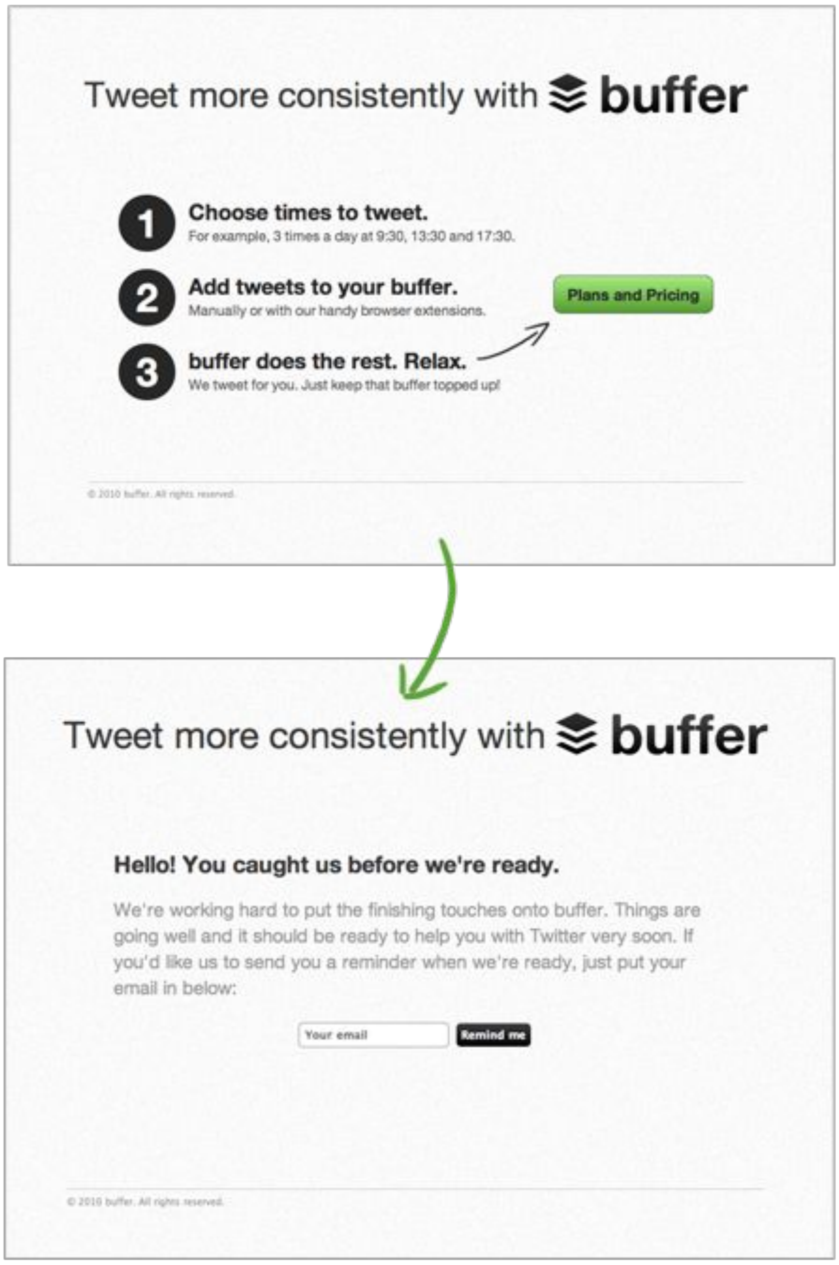
Smoke tests or *Fake Door* tests, test the demand for a product or feature that doesn't exist yet. The principle is "fake it until you make it" - users are shown a convincing option to try out the new product or feature or even to purchase it. Conversion rates at each step of the test will give us a sense of how desirable the product is. When users choose to opt-in, we inform them that the product or feature isn't ready yet, and often offer the option to join a waitlist — yet another test, as well as a way to generate leads. In addition to demand, smoke tests can help test pricing and value proposition.

Common forms of smoke tests include:

- Online ads and social media campaigns
- Landing pages describing the product and giving users an option to sign-up/buy.
- Buttons, links or other entry points in the UI that look like they'll "launch" the new feature
- Crowdfunding campaigns
- Physical storefronts, points of sale or conference booths
- Preview Videos that "demonstrate" the product experience and call users to sign up.

Caveats when using smoke tests:

- Smoke tests only test the top-most parts of your funnel. Having potential customers click on an ad or join a waiting list is very different from having them buy the product. Hence smoke tests only yield medium confidence.
- You want to expose only a fraction of potential users to a smoke test. Exposing all your users might reduce the effect of actually launching the feature or product, as well as undermine their trust in the product.



Tweet more consistently with buffer

1 Choose times to tweet.
For example, 3 times a day at 9:30, 13:30 and 17:30.

2 Add tweets to your buffer.
Manually or with our handy browser extensions.

3 buffer does the rest. Relax.
We tweet for you. Just keep that buffer topped up!

© 2010 buffer. All rights reserved.

Tweet more consistently with buffer

Hello! You caught us before we're ready.

We're working hard to put the finishing touches onto buffer. Things are going well and it should be ready to help you with Twitter very soon. If you'd like us to send you a reminder when we're ready, just put your email in below:

© 2010 buffer. All rights reserved.

Example: Buffer's fake landing page test source Buffer blog

Usability Tests

In *Usability Tests* we ask users to try using the product typically under the supervision and guidance of a researcher. We can test using static or interactive mockup created by a designer, a throwaway code prototype, or an actual interim version of the product. In all cases we'll try to make the user interface look reasonably realistic, but the functionality may be limited and we're likely to use canned data.

Usability tests can be conducted in a usability lab, in the user's home or workplace, in a neutral place such as a coffee shop, or remotely via video conferencing or a dedicated service. You can even go lighter and conduct *Guerilla Testing* (AKA *Hallway studies* or *Intercept Tests*), intercepting potential users in their natural environment (for example students in a university campus) and asking them to test the prototype or product on paper or on a mobile device.

It's important to plan your usability tests carefully - goals, number and profile of participants, what to test and how, and what to include in the test script. Usually a UX researcher or a designer will own planning and coordination, will recruit participants, carry out the test, then process and publish the results. However, usability tests (like most tests in this list) require cross-functional collaboration - product managers, designers, and engineers. It's good practice to have team members join the tests to take notes, or operate the manual parts of a test behind the scenes.

Human Operated Tests

A common way to simulate the functionality of yet-to-be built idea is to have humans do the work that the software would eventually automate.

- In *Concierge Tests* you have team members perform a service on behalf of the customer (often with the customer knowing it). The creators of Groupon started out by selling flash deals on their blog and then manually created the coupons and emailed them to customers.
- In *Wizard of Oz* tests you're typically conducting a usability test where the user is seeing a convincing user interface, while behind the scenes a human is doing the work. In the book *Build the right It*, innovation lecturer and coach, Alberto Savoia, explains how IBM tested a text-to-speech idea in the 1970s with a simple setup - a screen, a microphone and a hidden stenographer. Test participants saw how their words magically appear on the screen as text with high accuracy, but IBM researchers have learnt that they didn't love the feature - their voices become hoarse, they feared interrupting their colleagues, and they didn't like the fact that everything they entered into the computer could be heard by others. IBM had saved itself billions of dollars and years of hard work with this quick and simple test.
- In *Mechanical Turk* tests you crowdsource the work that eventually would be done by a machine-learning algorithm, to paid laborers.

Longitudinal User Study

Longitudinal user studies measure user reaction to the product over a period of time - weeks or even months. The test involves recruiting test participants and providing them with a version of the product they can use on their own. Often the participants are obliged only to try using the product once. After that, it's up to them if they wish to continue—they'll still get paid for the full test. During the test, we monitor usage, retention, and satisfaction as well as any qualitative feedback.

Longitudinal studies give us a sense of how much value the product provides and how much it draws people to use it again. A good result is seeing usage and satisfaction rates grow over time. Even better if you the participants clearly express disappointment when the test ends and ask when they can get the product back.

Early Adopter Programs

Early Adopter Programs give select potential customers early access to a new product in exchange for candid feedback and direct contact with the product team.Early Adopter Programs are especially important when developing product for medium or large businesses that are otherwise hard to test with. The participants should be true early-adopters — people who are willing to use an incomplete and not fully-tested product in their businesses or personal lives, in exchange for getting the benefit of the product ahead of everyone else, and helping shape the product.

The programs go by different names — early adopters, reference customers, trusted testers, alpha testing and more, and they usually don't require the participant to buy the product. The participants are more design partners than customers. However once the product is ready, they may become our first customers, as well as visible references and advocates.

It's important to keep the size of the program small so the product team can support the customers directly. During the program we prefer to have as few intermediaries as possible between the product team and the participants in the program.

Internal Company Tests

Before you subject your users and customers to a new product idea, it's best to test it on yourself and on your colleagues. Your coworkers will be more tolerant of bugs and missing functionality and can give you very early feedback on value, usability, and critical bugs. This process, often called dogfooding (short for "eat your own dogfood"), is a common practice in many large tech companies. Google, Microsoft, and Facebook religiously dogfood every product and feature long before they reach the market. Apple uses it extensively as it prefers to keep its products secret until they are announced.At Gmail, we often preceded dogfood with fishfood —testing a bare-bones version of the product with members of the team or with sister teams. As a side benefit, dogfooding and fishfooding are also powerful ways to build internal support for your idea and to flush out risks you might have missed.

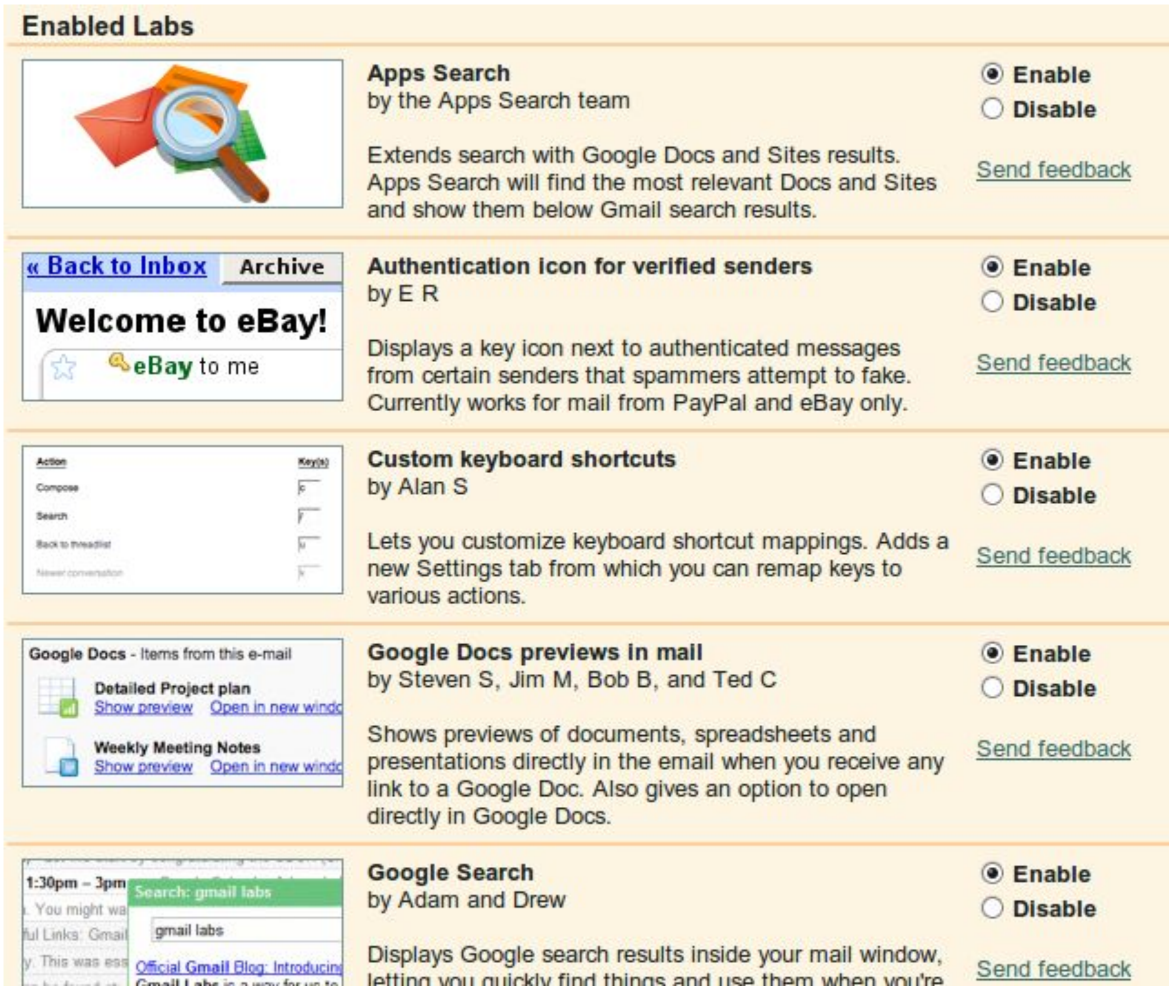
Dogfood participation may be the default for all company employees - at my first day at Microsoft I realized my Outlook email application was buggy. I was using the dogfood version by default. Other companies prefer to make dogfooding an opt-in. Google has an internal group of dogfood participants. Product managers will email the group and invite employees to join new dogfood tests in their products. The email should include criteria for participation and instructions on how to enable the test.

There are clear caveats to internal company tests:

- Not everything we build is genuinely useful for us or our colleagues. For example if you develop a medical product, you will not be using it unless you suffer from the condition for which it was designed.
- Even when the product is applicable, your coworkers are rarely exactly the same as your target market, and you get to use the product for free. The data and feedback they generate may be misleading. It's important to keep this in mind while analyzing dogfood results.

Labs

Labs are optional features in a product that users can choose to enable under the warning that the feature isn't fully complete, may not always function as expected, and that it also may be removed at any time. The Gmail team has tested many features as labs over the years. Some have "graduated" and became regular product features, including Preview Pane, Message Translation, and Custom Keyboard Shortcuts, while others were deprecated. In companies who used Gmail for work, administrators had the ability to control which labs were available to internal users and which weren't.



Enabled Labs

Apps Search
by the Apps Search team
Extends search with Google Docs and Sites results. Apps Search will find the most relevant Docs and Sites and show them below Gmail search results.
[✓ Enable] Disable
Send feedback

Authentication icon for verified senders
by E R
Displays a key icon next to authenticated messages from certain senders that spammers attempt to fake. Currently works for mail from PayPal and eBay only.
[✓ Enable] Disable
Send feedback

Custom keyboard shortcuts
by Alan S
Lets you customize keyboard shortcut mappings. Adds a new Settings tab from which you can remap keys to various actions.
[✓ Enable] Disable
Send feedback

Google Docs previews in mail
by Steven S, Jim M, Bob B, and Ted C
Shows previews of documents, spreadsheets and presentations directly in the email when you receive any link to a Google Doc. Also gives an option to open directly in Google Docs.
[✓ Enable] Disable
Send feedback

Google Search
by Adam and Drew
Displays Google search results inside your mail window, letting you quickly find things and use them when you're
[✓ Enable] Disable
Send feedback

Gmail's original labs

Labs create a safe zone for experimentation for both developers and users. Developers can test their projects earlier and get direct feedback from users as well as usage data. Users can get early access to features and help shape them.

Labs can also test just a part of the finished feature. For example when we developed the Tabbed Inbox in Gmail, we relied on a lab that only offered the classification (as Gmail labels) in order to improve the quality of our algorithms and measure whether users found them reliable enough. This illustrates another benefit of labs — getting early data prior to launch.

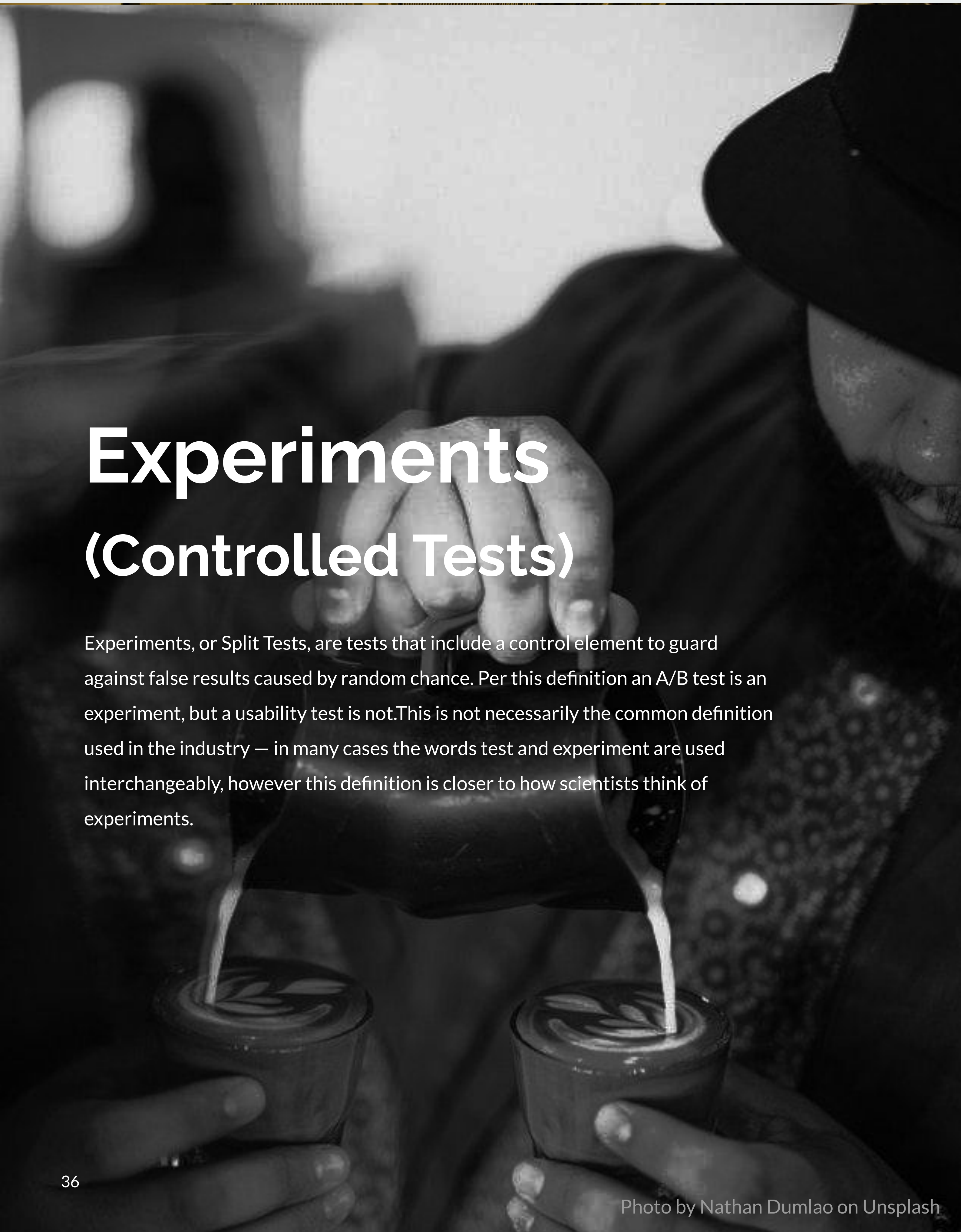
The evidence generated by labs is strong, but we must take into account that the users have self-selected to enable the lab and thus may not be representative of the larger population. It's important to set clear timeboxes for labs, by the end of which the lab is either deprecated or removed, otherwise some users will come to treat the lab as fully launched feature.

Beta Programs and Preview Versions

Beta programs and preview versions test near-final versions of the product, typically containing the final scope, but not fully bug-free or polished. The users consciously opt-in to test a yet-to-be-launched product. These late-stage programs act as a general dress-rehearsal before the launch, giving us access to mainstream users, higher loads and a lot more data. The programs may be open to all or by-invitation-only. Our goal is to test at scale so we are far less picky about who we let in compared to early-adopter/Alpha programs. We also don't offer the same level of direct-contact with the team. Support will likely be done via our usual support channels.

It's important to treat beta and preview as real tests rather than as mandatory milestones prior to launching. While the pressure to launch is high at this point, it's important to analyze the results critically - our goal is to learn. It's okay to stay in beta and iterate, as Google has done with many of its early products, or to revert a bad version altogether until we had time to fix what doesn't work right.

Gmail™
by Google™ BETA



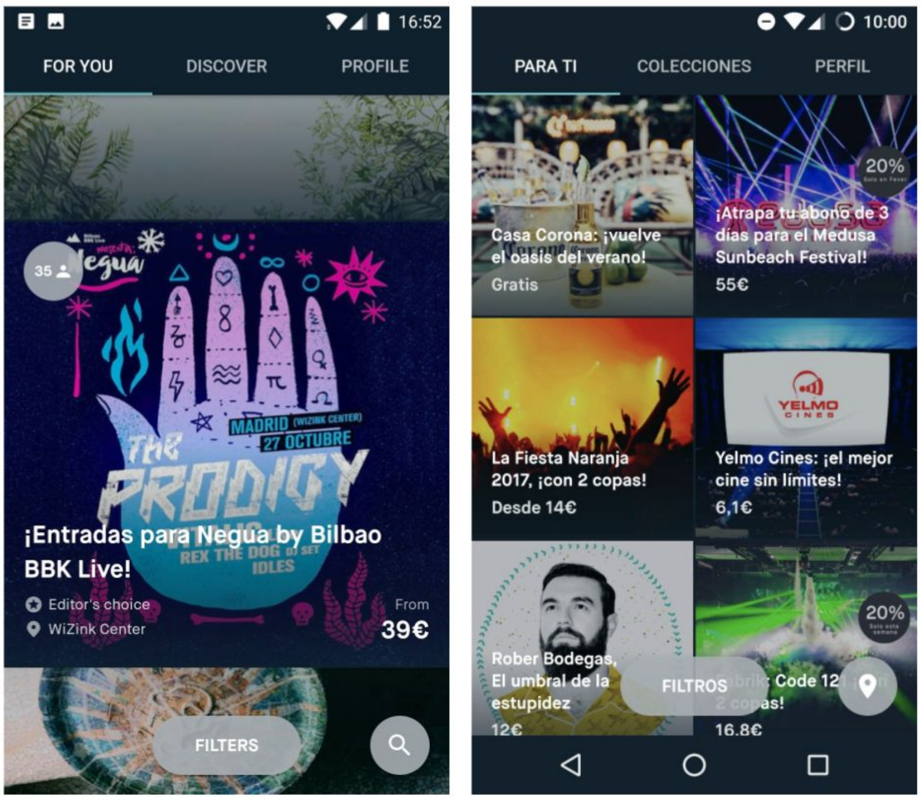
Experiments (Controlled Tests)

Experiments, or Split Tests, are tests that include a control element to guard against false results caused by random chance. Per this definition an A/B test is an experiment, but a usability test is not.This is not necessarily the common definition used in the industry — in many cases the words test and experiment are used interchangeably, however this definition is closer to how scientists think of experiments.

Photo by Nathan Dumlao on Unsplash

A/B Tests

A/B tests (AKA controlled experiments) compare user response to two versions of the product that are different in one single variable. Version A (control) is typically the current version, while version B (treatment) includes the change we wish to test — for example different button text or a new page design. Otherwise the two versions should be the same. As we expose two randomly selected groups of users of equal size to versions A and B of the product, we can measure differences in behavior, for example click-through rate on a button.


FOR YOU
DISCOVER
PROFILE
¡Entradas para Negua by Bilbao BBK Live!
Editor's choice
WiZink Center
From
39€
FILTERS
PARA TI
COLECCIONES
PERFIL
Casa Corona: ¡vuelve el oasis del verano!
Gratis
¡Atrapa tu abono de 3 días para el Medusa Sunbeach Festival!
55€
La Fiesta Naranja 2017, ¡con 2 copas!
Desde 14€
Yelmo Cines: ¡el mejor cine sin límites!
6,1€
Rober Bodegas, El umbral de la estupidez
12€
FILTROS
16.8€

Control - One event per screen   Experiment - Six events in grid view

Example: An A/B experiment carried out in event-finding app Fever. The control version A on the left shows only one event at a time on the screen, while the experiment version B on the right shows six events in a grid. The experiment version raised the number of events seen by users in by only 11% on average, and decreased CTR and overall revenue by 23%

A/B test are considered the most reliable validation technique because they can establish *causality* — a behavior change we measured was caused by the product change we made (and not say, seasonality, or a marketing campaign that ran at the same time).

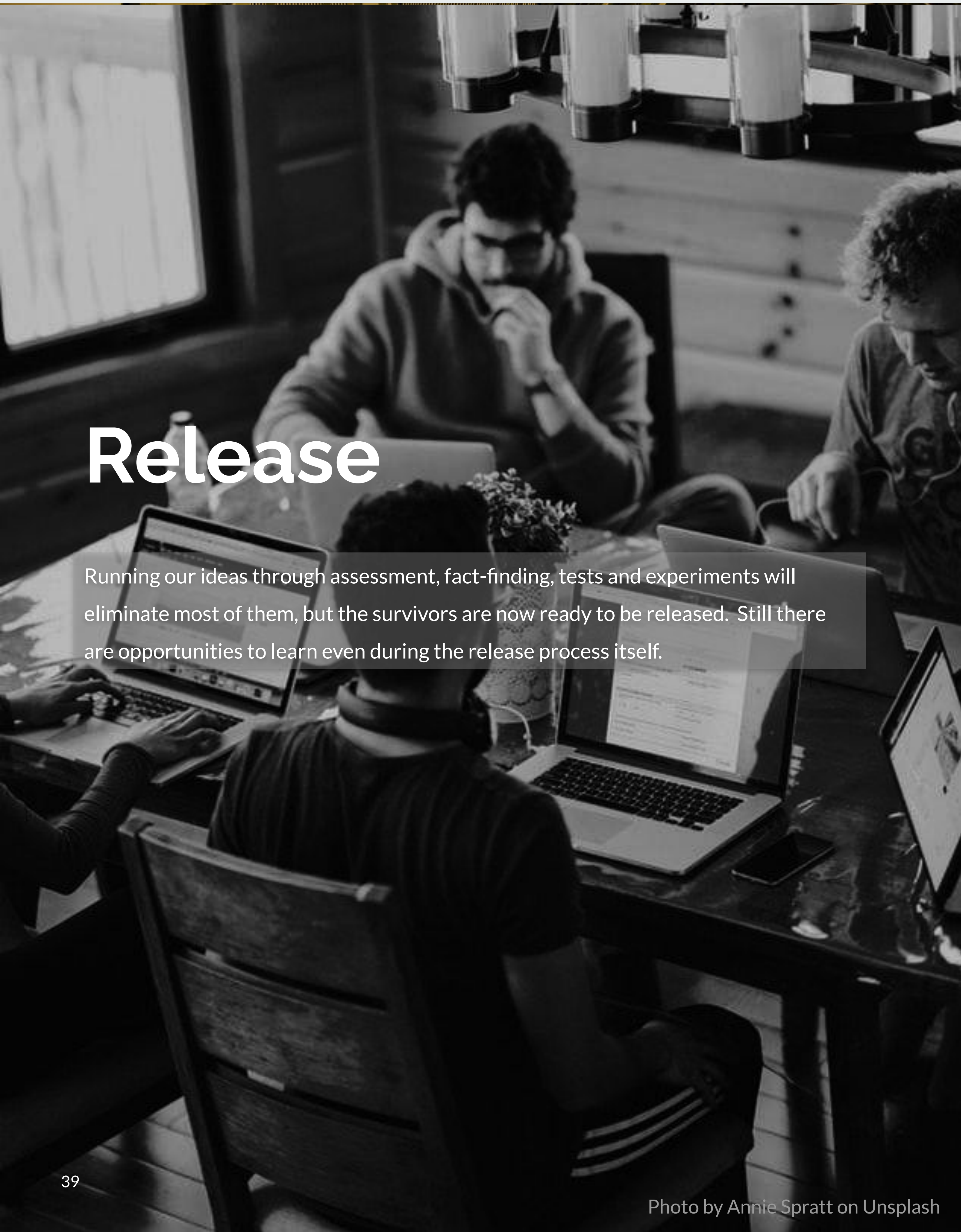
However to claim causality with high-probability we have to analyze the experiment using statistical tests. For example, using a 95% *statistical significance* test means that there's a 5% chance we would see the results we are seeing (or more extreme results) if there was no real change in the behavior of both groups. Statistical significance helps us determine causality, however, for various technical reasons, it does not exactly predict what percentage of our launched ideas will turn out positive (a common misinterpretation).

A/B/n tests

These are essentially the same as A/B tests, but we're comparing the control version A, to multiple experiment versions B, C, D… All the versions test the same variable - for example the color of a button. This type of experiment allows us to test more options, but also require most test groups and more data.

Multivariate tests

These split tests allow testing changes in multiple variables at the same time, for example the text on a button, its color, and its shape. We can test all the combinations or just a subset to see which combination yields best results. Multivariate tests require a lot more data and typically take longer to reach statistical significance.



Release

Running our ideas through assessment, fact-finding, tests and experiments will eliminate most of them, but the survivors are now ready to be released. Still there are opportunities to learn even during the release process itself.

Photo by Annie Spratt on Unsplash

Release Tests

Percent experiments

As we start gradually rolling out a product change to all our users, we may choose to stop at a specific rollout milestone — for example 25 percent, and conduct a large-scale A/B test, just to confirm that the results are consistent with what we've seen before.

Holdback experiments

As we're reaching full rollout of the change, we may choose to leave a small group of users with the old version to monitor the effects of the change over time.

Post Launch Data

If we do a good job during the validation period we are rarely surprised by what we find post-launch, but there's always a small chance we've missed something important. It's good to track to results of the launch over a number of months - usage, willingness to buy, bug reports, user feedback etc.

Interpreting Results

“Negative results are just what I want. They’re just as valuable to me as positive results. I can never find the thing that does the job best until I find the ones that don’t.”— Thomas A. Edison

Research doesn’t always yield clear results so it’s important to analyze the data we’ve gathered and ask what it means. You can do this alone, with team members, in formal reviews, or in regular meetings designed for this purpose (as Netflix does for example).

Here are some questions to ask:

Question 1: Are the results reliable?

The data we’ve collected may be corrupt, outdated, incomplete, or (in split tests) non-statistically significant. We may have biased the results somehow: sampling bias, functionality or usability bugs, leading questions, and anything else that may have interfered with the test. We may run into a seasonal or one-of events such as holidays or marketing campaigns. Sometimes the results simply don’t make sense — too good or too bad to be true, with no clear explanation.

In all these cases it’s best to stop here, redesign the test/experiment and rerun it.

Question 2: Are the results positive, negative or neutral?

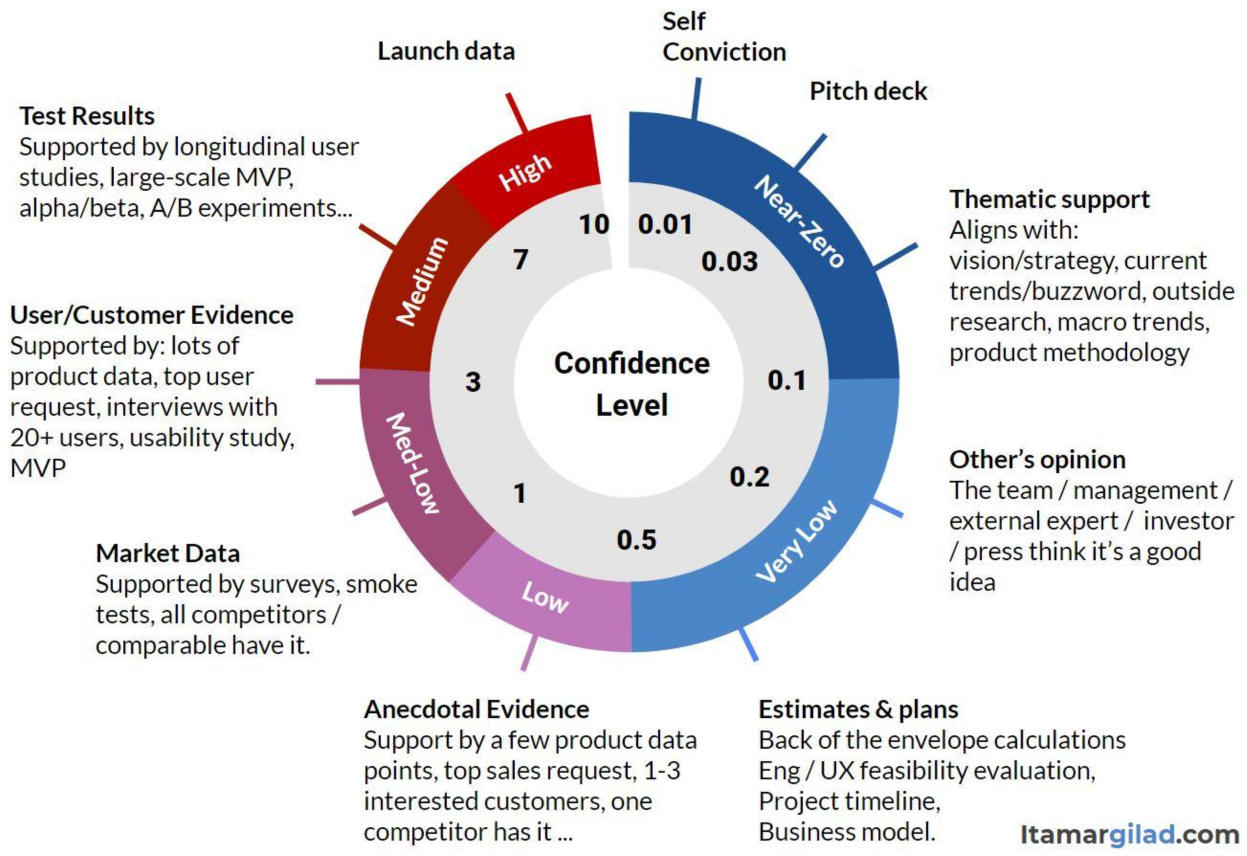
Data and test/experiment results are often not clearly positive or negative. If you conducted a poll and the idea only got 17% of the votes, is that bad or good? Is a measured uplift of 1% (statistically significant) in an A/B experiment clearly positive? It’s often a judgment call whether the data offers supporting or refuting evidence. We sometimes need to compare to other ideas and see what potential for improvement they bring. Your colleagues and managers can help in this discussion.

Question 3: How strong is the evidence?

Not all evidence is created equal. Having a few potential customers say they would use a new feature is great, getting customers to sign up to the early adopter program is a much stronger signal, and having them use the feature daily is stronger yet.

The strength of the evidence helps us determine how much confidence we should have in an idea, which in turn determines how much we're willing to to invest in it, and eventually, whether it's ready for a launch.

You can assess confidence using the Confidence Meter shown below, that translates evidence into a confidence score in the range of 0-10.



Self Conviction
Pitch deck
Thematic support
Aligns with: vision/strategy, current trends/buzzword, outside research, macro trends, product methodology
Other's opinion
The team / management / external expert / investor / press think it's a good idea
Estimates & plans
Back of the envelope calculations Eng / UX feasibility evaluation, Project timeline, Business model.
Anecdotal Evidence
Support by a few product data points, top sales request, 1-3 interested customers, one competitor has it ...
Market Data
Supported by surveys, smoke tests, all competitors / comparable have it.
User/Customer Evidence
Supported by: lots of product data, top user request, interviews with 20+ users, usability study, MVP
Test Results
Supported by longitudinal user studies, large-scale MVP, alpha/beta, A/B experiments...
Launch data
Near-Zero
Very Low
Low
Med-Low
Medium
High
0.01
0.03
0.1
0.2
0.5
1
3
7
10
Confidence Level
Itamargilad.com

The Confidence Meter

Deciding on the the next step

Once we've analyzed the results and determined they are reliable and clear, it's time to make a decision — given this new information, what should we do with this idea?

It's often good to define a simple policy like the one shown below:

Results	Confidence	Next Step
Supporting evidence	Higher	Develop the idea more and test it. If we've reached medium/high confidence: launch the idea.
Neutral results (no improvement)	No change	Park the idea OR change it (pivot) OR develop it further *
Contradicting evidence	Lower	Park the idea OR change it (pivot)
Unreliable results	No Change	Fix the test and rerun it.

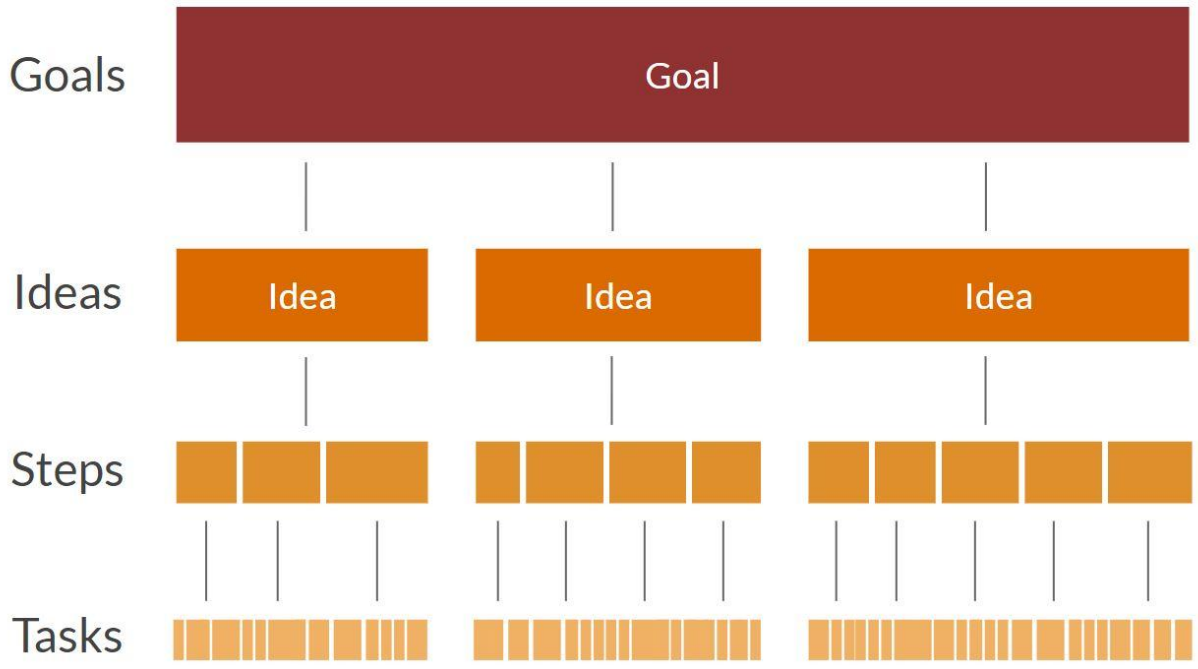
* The general rule is to **not** launch an idea that shows no measurable improvement because it does introduce new risks. However this is a judgement call. Sometimes we'll be willing to launch a no-measured-improvement idea because it does serve the product or business in way. For example we may launch a new visual refresh for our website, even if it has now affect, positive or negative, on any of our metrics, simply because we want to give our site a modern look.

Testing Ideas As Part of a System

Evidence-based product development requires more than a collection of validation methods. To really reap its benefits we need a system to guide us through the process of idea collection, validation and implementation. One such system is the GIST framework:

- **Goals** — Define what we wish to achieve. Without goals any idea may be valid.
- **Ideas** — Collecting ideas and ranking them so we can systematically choose what to validate first
- **Steps** — Iterating through idea validation, results collection and learning. [4]
- **Tasks** — The actual management of the work — Agile/Kanban is perfectly good here.

To learn more about GIST see this article [5].



The GIST Framework

About Itamar Gilad

Itamar is a coach, writer, and speaker specializing in product management and strategy. Prior to coaching he held senior product management and engineering roles at Google, Microsoft and a number of startups. Itamar publishes a popular product management newsletter where he shares articles, eBooks and templates.

.

NEWSLETTER

WORKSHOPS

KEYNOTES

CONTACT

References

[1] The Surprising Power of Online Experiments / Harvard Business Review -- Ron Kohavi and Stephen Thomke https://hbr.org/2017/09/the-surprising-power-of-online-experiments

[2] Assumptions Mapping https://precoil.com/resources/

[3] Frank Robinson's Minimum Viable Product Definition https://www.skmurphy.com/blog/2017/04/24/frank-robinsons-minimum-viable-product-definition/

[4] Choose better product ideas -- Itamar Gilad https://itamargilad.com/the-tool-that-will-help-you-choose-better-product-ideas/

[5] Why you should stop using product roadmaps and try GIST -- Itamar Gilad http://itamargilad.com/gist-framework/