TOOLKIT

Product Testing: A Guide to Getting Started

ACCION VENTURE LAB
LET'S SET THE SCENE.

You’re an early-stage fintech company with great potential and promise. Your user base is growing, and you want to continue building products that help you meet some ambitious growth and revenue goals.

The best and fastest way to get to product-market fit is to test, evaluate, and implement new ideas in a way that promotes innovation without sacrificing quality or time.

We’re here to help you do just that.
IN THIS TOOLKIT, WE’LL COVER...

• Building a test-friendly culture in your organization
• An experiment framework
• How to prioritize your tests
• Building the test plan
• Reporting results
• How to implement your findings

Note: Much of this toolkit is inspired and resourced from Optimizely’s Testing Toolkit, which can be found here: https://www.optimizely.com/resources/testing-toolkit/
Building a Test-Friendly Culture
Why Should I Test?

More Data, Less Intuition

Testing allows you to compare and contrast different user experiences – and show clear statistical differences in how they interact with your product.

Track the History of Your Product

You can measure your changes and see how they have affected key performance indicators across various customers at various times. And, you can even re-test ideas from the past to see how they fare!

Get Your Team Aligned with Product Changes

Open and frequent testing allows for more team members to suggest changes, allowing for more frequent opportunities to test – and encouraging a more agile approach to software development.

Testing is also becoming faster, more efficient, and cheaper than ever – making it a more reasonable choice!
Important principles for a good testing culture

**Measure Everything**
- Dashboards to measure and track KPIs
- Tools to track user behavior on the application
- Discuss key data points in team meetings and make decisions based on the data

**Determine Ownership**
- Responsibility for Product KPIs can be broken down within the Product/Engineering function
- Website/Marketing KPIs can be given to the Marketing organization
- Testing should be a joint effort between both

**Set Ways of Working**
- Deadlines and measuring/testing/reporting phases should be outlined (e.g. “cycles of work”)
- Testing Frameworks
- Prioritization and deployment

We’ll provide some resources to help with this
How To Test: Framing, Prioritizing and Executing
A Primer for Test Building

1. Set the Stage

What are we trying to solve? (ex. low user adoption of a particular feature)
What is the metric of success we measure against? (Revenue, DAU/MAU, churn)
What is the experience we want to test? (Layout of app features, landing pages, contacts)

2. Predict Outcomes

Proposed Change (specific and focused - ex. changing order of menu, button design, etc.)
Hypothesized effect ("[Change] will cause [metric] to [increase/decrease] by [%,#]"")

3. Test Details

Test Method
Qualitative and Quantitative data to use for testing
Select audiences to target
Duration needed to get proper sample size
Prioritize for impact and complexity

4. Test!
What to Test: Start Broadly, Break it Down

Problems to solve are usually macro-level and measurable...

...and can be broken down into various attributes of your product

“Lagging Revenue from in-app purchases”  
(Metric: revenue $)

Sales per visit  
Purchasing flow  
Optimal product mix

“Low adoption of a new feature”  
(Metric: DAU/MAU)

App layout  
Call to Action  
Feature overlap

We’ll follow an example of this as an exercise
Defining the attribute helps set boundaries around what to change – and potential outcome

**Our Attribute to Test:**

**App Layout**

**You Could Consider Testing...**

- **Action button placement**
- **User flow**
- **Content presentation**

**(REALISTIC!) Hypothesized Effect:**

“xx% improvement in feature visits”

**Set the Stage**

**Predict**

This layout can be tested in a variety of ways

- Search bar placement
- Company profile
- Engagement options
- User activity
- Nav bar content
Testing takes many forms – and vary in complexity

Two Popular Options: A/B Testing vs. Multivariate

A/B Testing

What it is: testing two variations of app or webpage design against one another

What’s it good for: testing a single variable or concept; bucketing your users into two distinct groups; smallest sample size needed

What it’s not good for: testing more complex design changes and user flow

Multivariate Testing

What it is: testing many variations of app or webpage design against one another

What’s it good for: can help with targeting large-scale changes in product design

What it’s not good for: small user samples (requires lots of traffic); a set of A/B tests would suffice for many design changes

A: Original layout
B: “Post” button moved to top navigation/search bar

v1: Original
v2: Moved search bar to bottom
v3: Swap of search and button bar
v4: Search and button bars on top
Then, determine the time required to get a good sample size for your test

1. Calculate Sample Pool
Your test pool will depend on several factors, such as:
- Daily Active Users (DAUs)
- % of users typically interacting with the item you’re testing
- % of DAUs that will receive the test

2. Minimum sample size
Your sample size will be based on a few statistical goals:
- Desired Confidence Level: how “confident” do you want to be in your tested outcome?
- Margin of Error
- Standard Deviation ($\sigma$)

3. Determine your duration
Test Duration must be at least

Note: if you have low user numbers, you may want to consider testing long enough for directional purposes

Also is helpful to account for day-to-day changes in DAU and other factors that may affect your testing pool

Go here for a handy calculator to determine test duration and more details on sampling
Finally, which test you decide to run will depend on the potential impact and the challenges of executing.

**Test Details**

**Determine the impact your change could make**
Assess a score of 1 (low impact) to 3 (high impact) for each criteria.

**Determine the ease in measuring the impact**
Assess a score of 1 (more difficult) to 3 (less difficult) for each criteria.

- What is the metric of success we measure against? (Revenue, DAU/MAU, churn)
- Secondary metrics to test
- Duration needed to get proper sample size
- Sample size needed to achieve significance
- Technical ease of the change (1= most complex) (3= least complex)

= Total Score (higher is better; min= 3 max= 9)

You can use the included Experiment Prioritization Framework to assess your various testing options.
Example: Let’s take our CTA proposal and test it

Test Type: A/B

A: Original Layout

B: “Post” button moved to top navigation bar

Test Details

Your Hypothesis: “20% improvement in feature visits”

Test Logistics and Statistics

- Daily app users: 3,000
- Typical visit rate: 20%
- % of users to test: 100% (half of users will get the variation)
- Confidence interval: 95%

Based on our calculation, you will want to run this test for **4 days** to get a proper sample.

We recommend tools like Google Analyze, Optimizely, or VWO to administer your test to your site/app visitors.
But also: you can use this for offline products!

As an example, if you’re an MSME lender, you can A/B test...

- **Loan terms** (APR, tenor)
- **Payback frequency** (weekly/every two weeks/monthly)
- **Servicing method** (agent, app-based, etc.)

**Establish a hypothesis:**
- “x% higher take rate”
- “x% lower default”
- “x% lower PAR”

**Some considerations**
- Ensure that there is an anchor point between test groups (similar credit score, branch, etc.)
- Testing will not provide instant results and may take more time to implement
- Ensure that testing outcomes are recorded effectively so they can be analyzed in a reasonable way and steers toward potential outcomes
Early companies should consider a few important realizations when testing

- **Patience is key – it takes time to test.**
  Be honest with your data – if you have fewer site/app visits, testing will be drawn out longer. **Avoid multivariate tests** if you’re looking to make quick decisions on which path to take.

- **Use a trusted testing tool.**
  Reputable testing tools will ensure **your data privacy laws are secured** while testing. Transactional data collected outside a secured environment that a testing platform provides runs the risk of privacy issues.

- **Test your ideas with the right users.**
  Ensure that you’re running tests with users that **fit into your target demographic for a product**. For example, testing an idea designed for high net-worth individuals with a lower-income user segment could steer users toward the wrong kinds of behaviors. (Testing tools can help with this!)
Reporting your Results and Some Considerations
Implementing your Results: Some Considerations

**Don’t Implement Everything.**
Your results may not be worth the time and effort to make a change – consider the cost, time to adjust, and whether the change moved the needle enough.

**Consider More Tests.**
Are there other product features or user behaviors you want to test against the metric you want to move? Explore other potential tests from your prioritization guide.

**On-Cycle Implementation.**
Don’t carve out space to implement changes – schedule them into your design and code sprints to ease pressure on your engineering/UX teams.

*If your A/B test resulted in a 5% improvement in visits, it’s probably not worth adopting.*

*Refer back to Slide 10 for the list of potential things you could test next – opportunities abound.*

*A simple button placement may seem simple, but it can take 2-3 sprints before there is space to implement it.*
Sharing results with your team is a crucial step

**What should we share?**

- What was tested?
- Why did we test it?
- What were our expected outcomes?
- How did we measure success?
- Who did we test it with/how long?
- What emerged as the “winner”?
- What are the implications/next steps?
- What should we test next?

**How do we share it?**

- Presentations (for stakeholders, determining next steps/go-ahead plans)
- Lunch and Learn sessions (for broader, org-wide lessons learned and updates)
- Technical/Engineering Blogs to showcase team’s data-driven focus
Additional Resources

- G2’s suggested A/B Testing tools: https://www.g2.com/categories/a-b-testing
- Experiment Prioritization Framework Worksheet