

Measure Up: Mobile Testing





Today, things get done, bought, and booked on the run. Which makes mobile experiences powerful revenue engines.

To make sure your mobile sites and apps are fueling the buyers' journey (and catapulting conversion), building, running, and analyzing tests is a must.

To kick things off, let's lay out the big differences between platforms – and why a separate strategy should be constructed for each.

Differences Between Mobile, Web, and Apps

There are two big distinctions between mobile, web, and app platforms: user tracking and interface dimensions. And although they might not sound like much, these variations are what drive the need for individual testing approaches.



Difference #1: Apps Cut Cookies

While web and mobile experiences use cookies to remember the pages and content shoppers have seen, apps use manual settings that allow shoppers to adjust experiences on their own.



Difference #2: Mobile & Apps Have Less Space

Unlike web, mobile sites and apps must deliver experiences that can be explored on small screens, with fingertips. That means buttons and input fields should be enlarged, information and images should be condensed, and key elements should be at the top of the page (always a good idea regardless of screen size or technology).

4 Steps to Shaping Mobile Experiences

Shape your mobile experiences to perform right from the get-go by following these four guidelines.

1 Know Your Users

Desktop users are often looking for ways to easily gather information and compare products. Only after these steps have been taken will they will feel comfortable continuing through the funnel to complete a purchase.

Visitors performing searches on mobile devices likely will not spend as much time comparing products, but rather are more likely to know what they want to purchase and are searching for further information to uphold their decision.

Backed with this knowledge, you can help guide your mobile users to relevant content quicker. This keeps them from getting frustrated or discouraged and leaving without the info they came to collect.

2 Build Experiences Based on Channel

As you scope your mobile experience, keep these considerations at the forefront:

- ▶ Call-to-action (CTA) copy should display the feature or benefit it will give a user (e.g., "See the Video," "Download the Paper," or "Send Me the Dress")
- ▶ CTAs should be larger than those on desktop, allowing visitors to easily navigate between pages
- ▶ Content should be both compelling and clear

3 Think Like a User to Analyze Data

When reviewing the performance of your mobile experience, walk a mile in a user's shoes.

Ask yourself:

- ▶ Based on what mobile users want, how do you think they were impacted by the changes made to the experience?
- ▶ Did your test elements help visitors navigate through the site to find relevant information?
- ▶ What were the drop-off points?
- ▶ Why do you think these drop-offs deterred visitors from staying on the mobile site?

4 Compare and Contrast Mobile and Desktop

If you run similar test campaigns on desktop and mobile, review the data side-by-side.

- ▶ Does the data for each lead to similar results? Or are you seeing major differences?
- ▶ Compare the results to the site objectives to expedite your learning and empower you to scope more dynamic campaigns in the future.

The Nuts & Bolts of Testing

As we get into the complex world of mobile testing, it's important to lay some groundwork.

The content that's served to a visitor is called an experience. And every experience can be broken down into a collection of elements. To explain this in further detail, let's use the typical landing page as an example.

Imagine our landing page includes a top navigation menu, a free-shipping message, and a large rotating banner above the fold. Each of those are individual elements that, when combined, make up an experience.

Performance testing allows you to compare the effectiveness of multiple elements, as well as the overall experience. This is beneficial for iterative development and ongoing site optimization, as well as brand new experiences and overhauling entire sites.

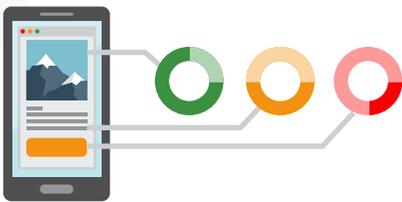
Types of Testing

To see if your mobile experiences have staying power, you've got to put them to the test. By tracking how visitors interact with various elements, you can identify patterns and make the call on which elements pack the most punch.

Now, let's look at two methodologies that make up the backbone of modern mobile testing: A/B (or split) testing and multivariate testing (MVT).



A/B testing pits a default experience against an alternative to see what performs best. A/B testing does not, however, report on the individual elements of an experience.



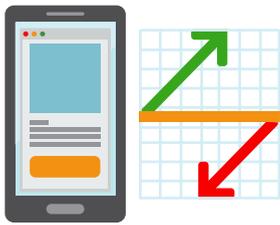
MVT (also called full factorial testing) compares various elements as well as different experiences against the default – and reports on both. For example, MVT can test multiple elements that are up for debate (e.g., hero image, headline and CTA) – as well as entire sites.

One-Tailed vs. Two-Tailed Tests



One-Tailed Test

A one-tailed test reports on the effect of a change in one direction, but not the other. For instance, after deploying a one-tailed test, you could ascertain if an element or experience being tested performs better than the default, although not whether it was equivalent or worse. One-tailed tests should only be used when you aren't concerned about missing an effect in an untested direction.



Two-Tailed Test

Two-tailed tests report on the effect of a change in both directions (both up and down) and enables you to determine if an alternative element or experience is better, equivalent, or worse than the default. Two-tailed tests serve as hard-hitting tools because they help mitigate the unintended side effects of altering an experience.

By accounting for all possible outcomes, this testing approach helps you report with confidence – and without the need for IT intervention. Two-tailed testing may mean more traffic and time, but you end up with way more reliable results.

Pick a Side: A/B or MVT

So, which testing type is best for you? Ask yourself these two questions – and start your testing off right.

Question #1:

Do you want to recreate your page from the ground up? Or are you looking to make incremental improvements?

If you want to make radical changes as part of a broader site or brand revamp, you need direction. For this, the most effective tool is an A/B test. A/B tests are extremely helpful for teams who need guidance on which trail to blaze.

If you want to measure iterative modifications rather than broad ones, you'll need to understand the performance of each element (as well as the performance of the overall experience). To avoid spurious seasonal effects from testing at different times, a multivariate test might be the best way forward.

Question #2:

Do have multiple incremental changes you'd like to test?

When you want to know the direct impact of a change, independently of any others, A/B gets the best results.

But when you want to compare the performance of more than one element, A/B can get tricky. Although we assume separate elements on a page aren't correlated, and that changing one (say, a banner) won't interfere with the effectiveness of another (like a social media button), this isn't always the case.

Statistical correlations are real. Ignore them and you risk mistakenly declaring the runner-up the champion. If you want to know the overall impact of multiple incremental changes on conversions or revenue, MVT is, hands-down, your best bet.

7 Ways to Speed Development Time for Your Tests

The test lifecycle can be long. And at times, arduous. With these tips, you can cut to the chase – and catapult your test through coding for quicker deployment.

1 Streamline Your Workspace

When objects (i.e. pages, actions, campaigns, scripts, and variants) are easier to find and navigate, existing functionality is easier to understand, and reusing existing objects and code is effortless.

For better organization:

- ▶ Come up with a naming convention for all objects
- ▶ Write clear descriptions for new objects
- ▶ Regularly audit your workspace to remove unused objects

2 Align Your Code to All Browsers

When adding new code to your testing workspace, remember that what you are writing will be running on all browsers – and make sure you're writing code that's compatible.

3 Follow Coding Best Practices

Help other developers save time and put their efforts to better use by:

- ▶ Thoroughly commenting
- ▶ Limiting line length
- ▶ Consistently indenting
- ▶ Using clear variable names

Also, be on the lookout for reusable code. You can stop wasted time in its tracks by keeping duplicate code in functions defined in global scripts and grouping similar code in the same script.

For example, keep all code related to calculating and collecting custom user attributes (such as gender or favorite product type) in one function in a global script. Then call that function anytime you need to update the user's attribute values and send them to a test.

4 Watch Out for Environment Differences

If you have to do your development work in a pre-production environment, learn the differences between the pre-production and live environment before development to avoid wasting time fixing bugs.

The most potent way to code a test is to use your testing platform's staging/sandbox environment, which will let you code against your live site's production code.

5 Track Default Production Site Changes

Changes to the default site are a problem, whether they happen during development or while the test is live. Always know what changes are being pushed to the production site and when they're arriving to avoid going back and to alter test code that has already been written.

6 Put Third-Party Integrations to Work

Some campaign metrics are best tracked using your existing analytics platform. Many major analytics packages can be integrated into existing platforms by sending aggregated data about which user population saw each of the test experiences, enabling you to analyze and compare these populations within one package.

Most metrics are best coded in your testing platform, but spare yourself hours of development work coding trivial metrics that will yield the same information as your analytics package. In addition, because of the nature of the metrics, some such as bounce rate are best monitored in an analytics platform.

7 Build for All Use Cases

To avoid bugs and collect clean data, all possible page experiences and scenarios should be accounted for when developing a test. Before you get going, brief departments in your organization on the campaign, so they can flag any use cases you didn't see coming.

ORACLE[®] Marketing Cloud

B2B and B2C marketers use Oracle Marketing Cloud, an integrated suite of marketing applications, to drive sales, brand and customer loyalty. Oracle Marketing Cloud leverages the industry's richest datasets and most adaptive machine learning to create an irresistible, consistent and connected experience wherever and whenever, however they choose to engage.

[Get yourself to oracle.com/marketingcloud](https://oracle.com/marketingcloud)