Primary and secondary metrics and monitoring goals

Skip Ahead

Primary metric
Secondary metrics
  Estimate time to statistical significance for multiple secondary metrics
Monitoring goals
Stats Engine approach to metrics and goals
  Revenue goals and skew correction with Stats Engine
Strategies for metrics
  Consider speed and impact
  Choose high-signal goals

This article will help you:

- Define testing goals for quick wins and long-term gains
- Set primary and secondary metrics and monitoring goals
- Create experiment and campaign goals and events that support your broader program metrics

Choosing the right metrics help you validate (or disprove) your hypothesis and ensure that you're making progress towards your overall business goals.

In experience optimization, metrics play different roles depending on where you set them and what you want them to tell you. Below, we walk you through each role type: primary and secondary metrics and monitoring goals. We also offer an Optimizely Academy course on understanding primary and secondary metrics.

The primary metric determines whether the test "wins" or "loses"—it tracks how your changes affect your visitors’ behaviors. Secondary metrics and monitoring goals provide additional information about your visitors’ behavior in the vicinity of your change and across your site. Monitoring goals are all goals and events that aren’t your primary or secondary metrics. They have minimal impact on the speed of secondary metrics and no impact on the speed of the primary metric.
Here are a few general tips for setting goals and events:

- **Focus on a direct visitor action** that is on the same page as the changes you made.

- **Consider how your changes affect other parts of your site.** Set goals and events to measure potential interaction effects so you know if your test is truly moves customers in the right direction.

- **Place different types of goals at different points in your funnel** to gather timely data about your visitors’ behavior.

If you’re wondering about the differences among goals, events, and metrics, check out [this article](#) for details.

Here are common metrics that programs track, based on revenue model.

### Primary metric

Optimizely allows you to set a primary metric for each experiment to determine its success. It’s the most important goal of the experiment and decides whether your hypothesis is proven or disproven.

In Optimizely, the primary metric will always achieve statistical significance at full speed, regardless of any other goals or events added. Stats Engine treats the primary metric separately because it’s the most important and tells you whether your hypothesis is supported.

Here’s [how to set up a primary metric](#) in Optimizely X.

In general, the more goals and variations you include in an experiment, the longer each will take to reach significance. For this reason, it's important to be mindful in distinguishing a primary metric from secondary metrics and monitoring goals. Stats Engine [corrects for false discovery rate](#) to help you make better business decisions.

When choosing a primary metric, ask yourself these questions:

**What visitor action indicates that this variation is a success?**

Often, the best path is to measure the action that visitors take as a direct result of this test.

**Does this event directly measure the behavior you're trying to influence?**

Many optimization teams automatically track revenue per visitor as the primary metric, but this isn't the best way to design a test. Top-level metrics like revenue and conversion rate are important, but the events involved are often far away from the changes made. If this is the case, your test may [take a long time to reach statistical significance](#) or end up inconclusive.

Use an action that's **directly affected by your change** to decide whether your test helped or hurt.
Does the event fully capture the behavior you’re trying to influence?

Consider whether your primary metric fully captures the behavior you’re trying to influence. What’s the best way to capture the change?

Imagine you’re testing the design and placement of an Add-to-Cart button. Your business cares about revenue, but it’s measured five pages down the funnel. You’re likely to devote a large amount of traffic to this test and you risk an inconclusive result.

You decide to measure clicks to the Add-to-Cart on product pages instead. It’s a primary metric that’s directly affected by the changes you made. And with a goal tree, you know that this metric rolls directly up to company goals.

Suppose you’re testing bulk discounts on your site. Your primary metric might be conversion rate, or it might be average order value (AOV). Neither metric fully accounts for the behavior you’re trying to affect.

The conversion rate could rise as customers are incentivized or decrease as customers wait to create large, discounted orders. AOV could rise as customers buy more in bulk or decrease as discounts take the place of full-price orders.

From this perspective, revenue-per-visitor is the best metric. It equals the conversion rate (how often customers purchase) multiplied by the AOV (how much they spend). It’s the best overarching goal in this test, where smaller goals may provide conflicting information.

Secondary metrics

Secondary metrics help you gather insights that are key to long-term success. In Optimizely X, the metrics that you rank from 2 to 5 are secondary metrics.

Secondary metrics track long-distance events and more ambitious metrics. End-of-funnel events like order value and order confirmation make excellent secondary metrics because they provide valuable information but are generally slower to reach significance. If you don’t make these long-term wins your primary metric, you don’t have to wait.

Secondary metrics are also useful for gaining visibility across the different steps of your funnel. For example, if you make a change to your product page and display shipping costs, your secondary metric might measure the change in drop-offs from the shipping page in your funnel. In general, use secondary metrics to learn when visitors drop off or navigate back to the home page and how these patterns compare between the original and variations.

Here’s a list of common secondary metrics:
<table>
<thead>
<tr>
<th>METRIC</th>
<th>REASON FOR TRACKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Searches submitted</td>
<td>See how many searches are submitted</td>
</tr>
<tr>
<td>Category pageview</td>
<td>Discover whether visitors navigate the site via Category pages</td>
</tr>
<tr>
<td>Subcategory pageview</td>
<td>Learn whether visitors reach Subcategory pages</td>
</tr>
<tr>
<td>Product pageview</td>
<td>Know the percentage of visitors who do or don't view a product during a visit</td>
</tr>
<tr>
<td>Add-to-Cart</td>
<td>Understand what percentage of visitors Add-to-Cart per test, category, product type</td>
</tr>
<tr>
<td>Shopping cart pageview</td>
<td>See how many visitors progress to the shopping cart</td>
</tr>
<tr>
<td>Checkout pageview</td>
<td>Understand how many visitors continue from the shopping cart to checkout</td>
</tr>
<tr>
<td>Payment pageview</td>
<td>Learn what percentage of visitors continue from checkout to payment</td>
</tr>
<tr>
<td>Conversion rate</td>
<td>Know what percentage of visitors ultimately convert or complete payment</td>
</tr>
</tbody>
</table>

**Note:**

To learn more about interpreting results across a funnel, [check out this article](#).

If you're using Optimizely X to test on a checkout page, you might need to [configure your site for PCI compliance](#).

**Estimate time to statistical significance for multiple secondary metrics**

Want to estimate how much longer it will take for multiple secondary metrics to reach statistical significance? Here's an easy back-of-the-envelope method.

In Optimizely's [Sample Size Calculator](#), fill out your baseline conversion rate and minimum detectable effect (MDE) as usual. For the statistical significance threshold, enter \( 100 - \frac{(100 - S)}{N} \), where \( S \) is your desired threshold (default is 90), and \( N \) is the number of metrics multiplied by variations other than baseline.

For example, if you are running an experiment with 2 metrics and 2 variations plus a baseline, at 90 significance, your secondary metric will require the number of visitors it takes to reach \( 100 - \frac{(100 - 90)}{(2*2)} = 97.5 \) significance with 1 goal and 1 variation.

This is an upper bound on the number of visitors you'll need on average, which means you'll likely see significance sooner.
Monitoring goals

Monitoring goals are all goals and events that aren’t your primary or secondary metrics. Like secondary metrics, monitoring goals help you gather insights that are key to long-term success, but they’re diagnostic and have minimal impact on the speed of secondary metrics and no impact on the speed of the primary metric.

Monitoring goals track whether your experiment is truly moving visitors in the right direction. Every time you create an experiment, you’re trying to optimize the user experience to improve a business outcome. But your change might also create adverse effects in another metric. Monitoring goals help you answer the question, “Where am I optimizing this experience, and where (if anywhere) am I worsening it?” Monitoring goals form a warning system that alerts you when you’re cannibalizing another revenue path.

For example, imagine that you show visitors more products on the product category page. With your primary metric, you find that people view more products as a result. Here are some other questions you might wonder at the same time, with the monitoring goal that can help you find out:

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>MONITORING GOAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are people more price-conservative when initially presented with more products?</td>
<td>Average order value</td>
</tr>
<tr>
<td>Are people actually buying more products?</td>
<td>Conversion rate</td>
</tr>
<tr>
<td>Are people frustrated and unable to find what they’re looking for?</td>
<td>Subcategory filters</td>
</tr>
</tbody>
</table>

Here is a list of common monitoring goals:

<table>
<thead>
<tr>
<th>GOAL</th>
<th>REASON FOR TRACKING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search bar opened</td>
<td>Learn what percentage of search bar interactions do not lead to submissions</td>
</tr>
<tr>
<td>Top menu CTR</td>
<td>Discover how often visitors navigate via the top menu per page or step in funnel</td>
</tr>
<tr>
<td>Home page CTR</td>
<td>See how often visitors exit to the Home page from any given page</td>
</tr>
<tr>
<td>Category page filter usage</td>
<td>Understand the frequency of filter usage</td>
</tr>
<tr>
<td>Product page quantity selection</td>
<td>Understand the percentage of visitors who interact with quantity selection</td>
</tr>
</tbody>
</table>
Stats Engine approach to metrics and goals

When you run an experiment with many variations and metrics, there’s a greater chance that some of them will give false positive results. In other words, it’s harder to declare winners when there are many variations and metrics.

Stats Engine uses false discovery rate control to address this issue and reduce your chance of making an incorrect business decision or implementing a false positive among conclusive results. As a result, Stats Engine becomes more conservative when you add more metrics to an experiment.

Here’s how Stats Engine prioritizes primary and secondary metrics and monitoring goals:

1. The primary metric, independent of all of the other metrics
2. Secondary metrics as an independent group of up to four metrics
3. All monitoring goals together as one group

This means that if you have 15 metrics attached to an experiment, Stats Engine will prioritize finding significance in the primary metric, then the secondary metrics, and finally the monitoring metrics.

Revenue goals and skew correction with Stats Engine

In general, Stats Engine works the same way for revenue-per-visitor goals as it does for other goals. You can look at your results any time and get an accurate assessment of your error rates on winners and losers, as well as difference intervals on the average revenue per visitor (RPV).

However, when interpreting your results for RPV goals, there are some differences you should be aware of.

Testing for a difference in average revenue between a variation and baseline is more challenging than testing for a difference in conversion rates. This is because revenue distributions tend to be heavily right-tailed, or skewed. This skewness impedes the distributional results that many techniques rely on, including t-tests and Stats Engine. The practical implication is that they end up having less power or are less able to detect differences in average revenue when those differences actually exist.
Optimizely’s Stats Engine regains some of this lost power through **skew correction**. Skew corrections were specifically designed to work well with all other aspects of Stats Engine.

Thanks to skew correction, confidence intervals for continuously-valued goals are no longer symmetric about their currently observed effect size. The underlying skewness of the distributions are now correctly factored into the shape of the confidence interval. Additionally, detecting differences in average revenue is more reasonable for the types of visitor counts that Optimizely customers regularly see in A/B tests.

---

### Strategies for metrics

Use the strategies described in this section to help you decide what metrics to use for your experiments.

#### Consider speed and impact

Think of your primary metric in terms of distance. In a funnel, the most immediate effects are directly downstream from the changes you made. The closer an event is to the change, the louder the signal and the bigger the measurable impact. As you move downstream, the signal starts to fade as visitors from different paths and motivations enter the stream. At the end of the funnel, the effect may be too faint to measure.

Remember, all other things being equal, **metrics that have a lower conversion rate require more visitors to reach statistical significance**. Events that are further from the page you’re testing will have lower improvement in conversion rates due to your variation as visitors enter from different paths, leave the site before they convert, and more. If this is the case, your test will take longer to reach significance.

Instead, **consider setting a primary metric on the same page as your change**. The impact of your change will be picked up immediately, so you’ll quickly find a winning variation. Quick wins help generate credibility and interest in your testing program and provide fast, reliable insights about how your visitors behave. By focusing on small, grounded wins, you build a testing program that’s data-rich and can quickly iterate on the insights it generates.

Ambitious, program-level events like revenue and conversion rate make excellent secondary metrics and help keep your program focused on long-term success.

#### Choose high-signal goals

As we [mentioned above](#), Optimizely's Stats Engine reacts to the number of goals or events and variations in your experiment to align statistical significance with your risk in making business decisions from experiments. We also mentioned that significance takes longer to achieve when there are more goals or events and variations in an experiment and that your primary metric is exempt from this slowdown. Here, we’ll add to the story.

Adding more goals or events and variations to your experiment increases your chance of implementing a falsely significant result with traditional statistics, and this is what Stats Engine corrects ([here's a detailed explanation](#)). However, not all goals and events are equal. High-signal goals and events—those you believe will be most affected by your variations—are less likely to contribute to false discoveries. This is because high-signal goals and events are usually less noisy, so it is easier to tell if your variation is having an effect on them.
One analogy to think of is to consider experimentation with multiple goals or events and variations as trying to pick needles of true differences from a haystack of noise. It is easier to find a large (high-signal) needle in the haystack than a small (low-signal) needle.

Similarly, with Stats Engine, the more high-signal goals in your experiment, the faster all your secondary metrics and monitoring goals will reach significance. If speed to significance is a concern for your organization, consider limiting the number of non-primary metrics in your experiment and focusing on goals or events that you believe are related to your variations.

Of course, you are free to add many metrics to your experiments. The strength of Stats Engine is that you will not be exposed to higher error rates, but the cost of broad, undirected exploration is longer time to significance.