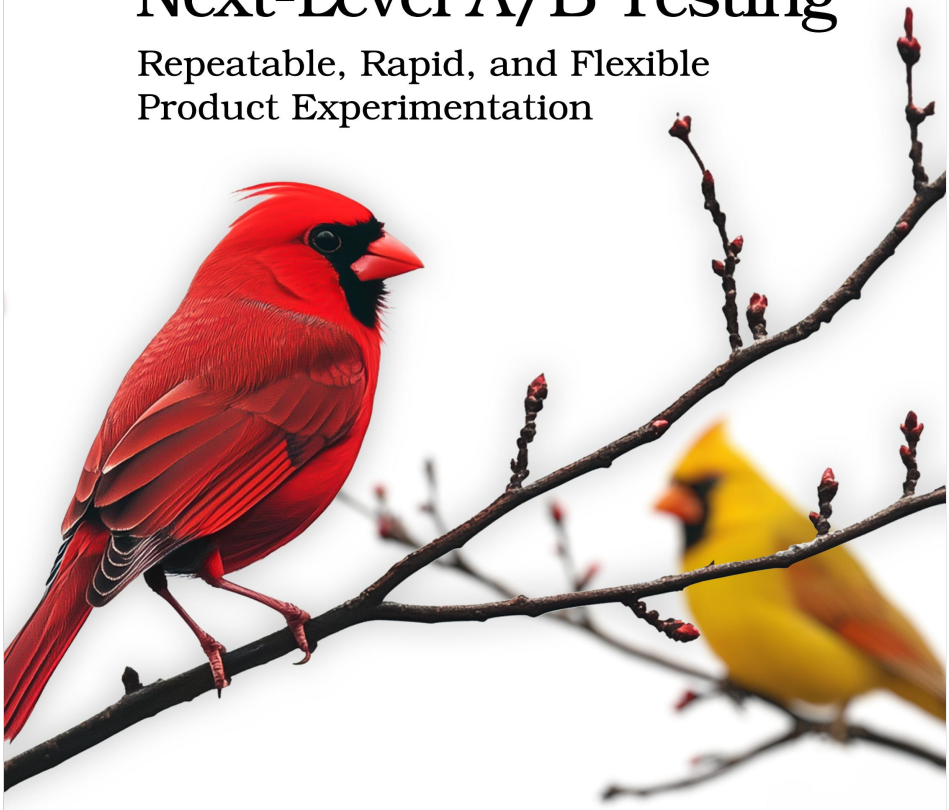
The Pragmatic Programmers

Next-Level A/B Testing

Repeatable, Rapid, and Flexible Product Experimentation



Leemay Nassery
edited by Vanya Wryter

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Peeking and Ending Tests Early with Sequential Testing

The first strategy we'll explore is the sequential testing paradigm. Unlike the most traditional testing strategy, more formally referred to as a fixed-horizon or fixed-size test, sequential testing does not require committing to a fixed sample size for the control and test variants. Sequential testing promotes peeking, which means checking your test results early and continuously.

See the following image to compare the classical fixed-horizon experiment design versus the sequential experiment.

	Sequential Testing	Fixed Size Testing
Is the sample size set in advance?	No	Yes
Can you peek at the data?	Yes	No
Can you end the test early?	Yes	No

With sequential testing, you have the flexibility to analyze data as it is collected while the experiment is still active. This allows for interim analyses, where statistical checks are performed to avoid false interpretations and maintain the integrity of the results. The sooner you can determine whether to launch, improve, or discontinue the feature being tested, the quicker you can conclude the experiment and free up resources for new tests.

There are several advantages to peeking at your test early by running a sequential test. First, you can end an experiment early if it's degrading key metrics to reduce the metric hit on a longer timeline. Second, you can end an experiment early if it's performing well from a metrics standpoint by launching the change to all users to continue to affect metrics positively but at a larger scale. Third, you can use the platform's testing resources more efficiently by reducing the time and sample size.

On the other hand, there are several disadvantages to running a sequential test. First, novelty effect, when a new feature is introduced to users and engagement increases but then fades after the initial exposure, may be

harder to detect if you end an experiment early because the insights look promising. Second, looking at test results early could lead to premature decisions, potentially favoring a version of the new the feature that appears successful initially but doesn't perform well over time.

The sequential testing strategy may lend itself to increasing your experimentation rate because you're ending tests early by peeking at the results compared to the traditional fixed-horizon test. For instance, if you stop a test early because you observed postive gains your success metrics, you can allocate users to another experiment. If the positive gain is on a business metric, such as revenue, imagine the importance of enabling the feature earlier rather than later. You want to launch a feature that increases revenue sooner rather than later from an economic standpoint.

The sequential testing strategy can increase your testing speed by allowing you to end tests early based on interim results, unlike traditional fixed-horizon experiments. For example, if you observe positive gains in your success metrics and stop a test early, you can quickly start another experiment with the freed-up users. This approach is especially beneficial for experiments focused on revenue or other business metrics, as launching a feature that boosts revenue sooner is economically advantageous.

There are multiple subtypes of sequential testing, such as group sequential testing and always valid inference. For more details on these two types of sequential testing, consider reading a paper titled "Group Sequential Designs: A Tutorial"[1] and "Anytime-Valid Linear Models and Regression Adjusted Causal Inference in Randomized Experiments."[2]

Now that you're familiar with sequential testing, let's define the multivariate testing strategy.

Evaluating More with Multivariate Tests

Multivariate testing involves presenting multiple changes to users simultaneously within a single experiment. This typically includes several test variants, each evaluating different combinations of these changes to identify the most effective configuration.

Advantages to multivariate testing include discovering the best design for a page holistically as you're encapsulating more than one change into one test

1. https://psyarxiv.com/x4azm/
2. https://arxiv.org/abs/2210.08589

variant, and reducing the number of distinct experiments by combining multiple changes at once to evaluate collectively.

Advantages of multivariate testing include:

1. Holistic Design Optimization: By testing multiple changes within a single variant, you can discover the best overall design for a page.
2. Efficiency: It reduces the number of separate experiments needed by combining multiple changes into one test, allowing you to evaluate them collectively.

As for the most common disadvantages of running multivariate testing, consider the following:

1. Higher Resource Requirements: It needs more users and time to evaluate the combinations tested within different variants.
2. Complexity in Analysis: It can be difficult to isolate which specific change affected your metrics.

Therefore, it's essential to clearly articulate the goal of the experiment through a well-defined hypothesis when employing the multivariate testing strategy.

If your goal is to understand a combination of changes that ladder to a holistic product vision with less concern about attributing metric effects to specific features, multivariate testing is an excellent strategy to use. However, if your goal is to understand the specific impact of a particular feature, strategies like isolated or sequential testing may be more effective than multivariate testing.

Let's explore isolated testing further in the following section.