Experimentation

Want Your Company to Get Better at Experimentation?

by Iavor Bojinov, David Holtz, Ramesh Johari, Sven Schmit and Martin Tingley

From the Magazine (January–February 2025)

Jamie Chung/Trunk Archive

Summary. For years, online experimentation has fueled the innovations of leading tech companies, enabling them to rapidly test and refine new ideas, optimize product features, personalize user experiences, and maintain a competitive edge. The widespread... **more**

Leer en español

For years online experimentation has fueled the innovations of leading tech companies such as Amazon, Alphabet, Meta, Microsoft, and Netflix, enabling them to rapidly test and refine new ideas, optimize product features, personalize user experiences, and maintain a competitive edge. Owing to the

widespread availability and lower cost of experimentation tools today, most organizations—even those outside the technology sector—conduct online experiments.

However, many companies use online experimentation for just a handful of carefully selected projects. That's because their data scientists are the only ones who can design, run, and analyze tests. It's impossible to scale up that approach, and scaling matters. Research from Microsoft (replicated at other companies) reveals that teams and companies that run lots of tests outperform those that conduct just a few, for two reasons: Because most ideas have no positive impact, and it's hard to predict which will succeed, companies must run lots of tests. And as the growth of AI—particularly generative AI—makes it cheaper and easier to create numerous digital product experiences, they must vastly increase the number of experiments they conduct—to hundreds or even thousands—to stay competitive.

Scaling up experimentation entails moving away from a data-scientist-centric approach to one that empowers *everyone* on product, marketing, engineering, and operations teams—product managers, software engineers, designers, marketing managers, and search-engine-optimization specialists—to run experiments. But that presents a challenge. Drawing on our experience working for and consulting with leading organizations such as Airbnb, LinkedIn, Eppo, Netflix, and Optimizely, we provide a road map for using experimentation to increase a company's competitive edge by (1) transitioning to a self-service model that enables the testing of hundreds or even thousands of ideas a year and (2)

focusing on hypothesis-driven innovation by both learning from individual experiments and learning *across* experiments to drive strategic choices on the basis of customer feedback. These two steps in tandem can prepare organizations to succeed in the age of AI by innovating and learning faster than their competitors do. (The opinions expressed in this article are ours and do not represent those of the companies we have mentioned.)

The Current State

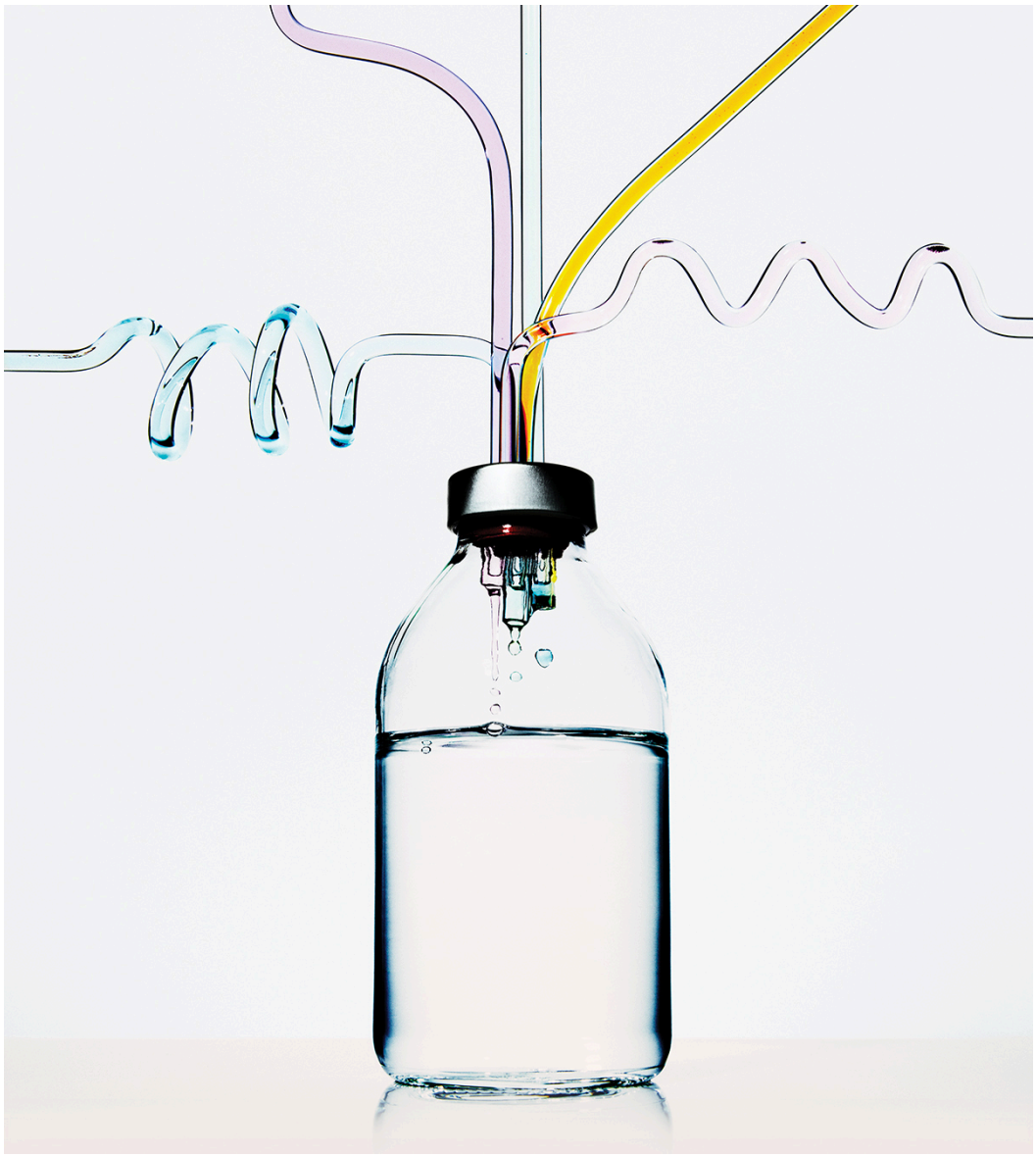
The basics of experimentation are straightforward. Running an A/B test involves three main steps: creating a challenger (or variant) that deviates from the status quo; defining a target population (the subset of customers targeted for the test); and selecting a metric (such as product engagement or conversion rate) that will be used to assess the outcome. Here's an example: In late 2019, when one of us (Martin) led its experimentation platform team, Netflix tested whether adding a Top 10 row (the challenger) on its user interface to show members (the target population) the most popular films and TV shows in their country would improve the user experience as measured by viewing engagement on Netflix (the outcome metric). The experiment revealed that the change did indeed improve the user experience without impairing other important business outcomes, such as the number of customer service tickets or user-interface load times. So the Top 10 row was released to all users in early 2020. As this example illustrates, experimentation enables organizations to make data-driven decisions on the basis of observed customer behavior.

Barriers to Scaling Up Experimentation

Data science teams often lead the adoption of online experimentation. After initial success, organizations tend to fall into a rut, and the returns remain limited. A common pattern we see is this: The organization invests in a platform technically

capable of designing, running, and analyzing experiments. Large technology companies build their own platforms in-house; others typically buy them from vendors. Although these tools are widely available, investing in them is costly. Building a platform can take more than a year and usually requires a team of five to 10 engineers. External platforms generally cost less and are faster to implement, but they still require dedicated resources to be integrated with the organization's internal development processes and to gain approval from legal, finance, and cybersecurity departments.

After the initial investment, leaders who sponsored the platform (usually the heads of data science and product) face pressure to quickly demonstrate its value by scoring successes—experiments that yield statistically significant positive results in favor of the challenger. In an attempt to avoid negative results, they try to anticipate which ideas will have a big impact—something that is exceptionally difficult to predict. For example, in late 2012, when Airbnb launched its neighborhood travel guides (web pages listing things to do, best restaurants, and so on), the content was heavily viewed, but overall bookings declined. In contrast, when the company introduced a trivial modification—the ability to open an accommodation listing in a new browser tab rather than the existing one, which made it easier to compare multiple listings—bookings increased by 3% to 4%, making it one of the company's most successful experiments.



Jamie Chung/Trunk Archive

Motivated to turn every experiment into a success, teams often overanalyze each one, with data scientists spending more than 10 hours per experiment. The results are disseminated in memos and discussed in product-development meetings, consuming many hours of employee time. Although the memos are broadly available in principle, the findings they contain are never synthesized to identify patterns and generalizable lessons; nor are they archived in a standardized fashion. As a result, it's not uncommon for different teams (or even the same team after its members have turned over) to repeatedly test an unsuccessful idea.

Looking to increase the adoption of and returns from experimentation, data science and product leaders tend to focus on incremental changes: increasing the size of product teams so as to run more experiments and more easily prioritize which ideas to test; hiring additional data scientists to analyze the increased number of tests and reduce the time needed to execute on them; and instigating more knowledge-sharing meetings for the dissemination of results. In our experience, however, those tactics are unsuccessful. Managers struggle to identify which tests will lead to a meaningful impact; hiring more data scientists provides only a marginal increase in experimentation capacity; and knowledge-sharing meetings don't create institutional knowledge. These tactics may appear sensible, but they end up limiting the adoption of experimentation because the processes they establish don't scale up.

Democratizing Experimentation

To achieve enterprise-wide experimentation for data-driven decisions, companies have to transition to a self-service approach: empower all employees on the product, marketing, engineering, and operations teams to test changes small and large and then learn from and act on outcomes. That means embedding some important functions in the platform and redesigning the data scientists' jobs.

The platform. The data science organization (data scientists, data engineers, and software engineers) should ensure that the platform contains the following features, whether it is built internally or purchased.

A simple, easy-to-understand interface. Airbnb had such a system, which enabled a single engineer to implement and test the feature that opened accommodation listings in a new tab.

The ability to automatically impose statistical rigor. Tasks such as determining the appropriate duration for a particular type of experiment and the criteria for deciding whether the results are significant should be automated using historical data.

Embedded experimentation protocols. Instructions should provide default settings for most aspects of standard experiments, such as decision-metric selection. These protocols allow users to design and launch experiments with minimal input from data scientists.

Automated rollbacks. These are quantitative criteria that act as trip wires to stop an experiment if its impact is too negative—for example, a significant drop in the number of daily active users of a social media site. The impact is measured using guardrail metrics—secondary measurements that ensure that while you're focused on improving one outcome, you don't unintentionally harm other important areas such as user experience, revenue, or system stability. When a vast number of experiments are running concurrently, such a feature is vital.

An AI assistant that provides easy-to-understand explanations of complex concepts. This core element can simplify the design and analysis of experiments, making the process accessible even to novice users.

Data scientists' role. In addition to setting up the platform, data scientists should be responsible for training employees, creating the materials for that training, and holding office hours to answer complex questions after everyone is up and running. The time they spend on most tests will drop to nearly zero because they will no longer be involved in execution or analysis. (They will still be involved in novel tests, such as the first in a new product space, and will be called in when results are challenging to interpret. But those are the exceptions.) Thus they can focus on projects of greater impact that leverage their unique expertise: for example, developing new statistical methods for analyzing complex

experiments and analyzing company data in light of past test results to identify new possibilities for product initiatives.

Preparing the Organization

In organizations that have not adopted experimentation, product teams are generally evaluated according to whether they launch new products. When they start experimenting, too often the criterion becomes the number of "successful" experiments run. Unfortunately, that makes employees risk-averse, so they run too few experiments. Scaling up experimentation, therefore, requires changing incentives. Companies should evaluate employees on the basis of the overall performance of the business unit and the organization, not the outcome of individual tests.

That shift will encourage a far wider range of employees to generate and test as many ideas as possible, increasing their chances of discovering breakthroughs that enhance performance. But it will also result in testing potentially higher-risk ideas with less oversight from experienced data scientists—something that can make people hesitant to run experiments. As we mentioned, one solution is to embed guardrails (quantitative criteria that act as trip wires) in the platform. Another is to roll out new features or changes in phases—a practice common among the largest tech firms. For example, updates to mobile apps from the Apple App Store and the Google Play Store are released that way to reduce risk.

Hypothesis-Driven Innovation

As organizations adopt and scale up experimentation throughout the enterprise and transition to an incentive model that rewards overall business impact, product leaders should be able to extract significantly more value by focusing on understanding the *why* behind test results. That requires managers to use experimentation for more than making data-driven decisions—such as whether a particular change is better than the status quo

—by hypothesizing *why* that is so. The experiment allows them to test the theory; by considering additional metrics, they can understand the mechanism that drove the result. Crucially, a focus on *why* fuels more customer-centric innovation, because feedback—gathered through experiments—is consulted not only to choose between the variant and the status quo but also to determine the next experiment and the overall product direction.

Netflix's Top 10 experiment, for instance, began with a clear hypothesis: The Top 10 row would help members find content to watch by tapping into an innate desire for shared experiences and conversations. That would increase member joy and satisfaction, as measured by increased member engagement. In addition to tracking overall engagement, the experiment monitored metrics such as where members found content (Search, My List, various rows on the home page) and how they interacted with the titles showcased in the Top 10 row. (Those titles were also available in the status quo experience but in a different location.) The additional metrics demonstrated how members changed their behavior in response to the new row. For example, because Netflix aims to connect members with the best content for them directly from the home page, an increased use of Search in response to the Top 10 row would indicate that the home page had not been delivering on that goal. That information would be used to design a subsequent test.

Once an organization is running hundreds or thousands of experiments a year, however, it becomes impossible to review every one of them in dedicated memos and meetings. Organizations should therefore shift their focus from analyzing individual experiments to analyzing, discussing, and learning from groups of related experiments, such as those concerning the search function or product-details pages that provide pictures, specifications, reviews, and other information. We refer to such

efforts as *experimentation programs*. This shift is the key to unlocking significant additional value from experimentation. When experiments are considered in this way, an organization can embrace more-efficient, hypothesis-driven innovation practices that build on prior tests to inform future ones. Experimentation programs also encourage product teams to break complex ideas down into small, testable hypotheses, making it easier to adapt the direction of a product to customer demands.

Experimentation Programs

Once an organization has become competent at learning across experiments, the next step is to compare results across experimentation programs, which makes it possible to evaluate the relative performance of various product areas and identify potential investment opportunities. Consider an e-commerce platform that has multiple features designed to help shoppers find the right product, two of which are the search function and the product-details page. The business would most likely have one experimentation program for search and another for product pages.

Jamie Chung/Trunk Archive

Now suppose that changes to the ranking algorithm used in a search engine generated positive but diminishing returns, as measured by the effect on sales in successive experiments. Meanwhile, all but one of the tests on the product-details page consistently showed small negative effects on sales, and that one exception produced large positive effects. One big “win” for the product-details page amid a number of unsuccessful tests suggests that the company doesn’t yet understand what aspects of product description resonate most with customers. Additional resources should be devoted to that experimentation program. Meanwhile, the diminishing returns on search-ranking experiments suggest a mature search-engine algorithm; leaders should consider either exploring vastly different approaches—such as an AI chatbot—or shifting resources to other areas for experimentation, such as product-details pages.

A Knowledge Repository

Learning across experiments at scale requires creating a knowledge repository—a system designed to store, categorize, and organize experiment results (including effects on sales and other key metrics, hypotheses about impacts on customers, and so on)—and making the information in it accessible to data scientists, product managers, and leadership. A repository allows the organization not only to track the state of any experimentation program but also to spread learning across the enterprise, which is crucial for hypothesis-driven innovation when a company is running a huge number of experiments each year.

A knowledge repository should perform four key functions: (1) It should make it possible to group experiments into programs. Many organizations would most likely group them by feature (such as search engine or product details) or business unit (such as marketing or customer support). (2) It should store and track the KPIs (quantity sold, revenue, conversions, and so on) that are important across the business. That will allow the impact of various experiments and experimentation programs to be compared on common terms. For example, most of Netflix's

experiments are designed to improve one of a handful of KPIs, such as engagement. (3) It should host all documents related to each test, mapping them to the experimentation program to ensure that all learnings are centrally available. (4) Most important, it should enable all employees to easily extract insights. Dashboards that track the performance of experimentation programs (such as the number of experiments run, the number of feature changes rolled out to the entire user base, and the cumulative impact of experiments on users over the previous quarter) are a great starting point. However, a more dynamic access point is an "assistant" powered by generative AI that can answer complex questions about past experiments.

...

Leading tech organizations use experimentation to innovate and improve performance rapidly by testing all ideas—not just carefully vetted ones or only the big ones. Moreover, learnings from those experiments (often gleaned from combining results across similar experiments) generate new ideas for testing. Experimentation can be scaled up only by democratizing access to tools, aligning incentives with improvements in long-term outcomes, and enabling employees to easily view, compare, and synthesize the results of experiments both within and across experimentation programs. Thanks to modern data tools and advances in AI, becoming expert in experimentation is now within reach for many more organizations. Given that the same AI advances are reducing the cost of coming up with, testing, and building innovative product variants, leaders must turn what is possible into a reality in their organizations.

A version of this article appeared in the January–February 2025 issue of *Harvard Business Review.*

Iavor Bojinov is an assistant professor of business administration and the Richard Hodgson Fellow at Harvard Business School. He is also a faculty affiliate of Harvard's statistics department and the Harvard Data Science Initiative.

David Holtz is an assistant professor in the management of organizations and entrepreneurship and innovation groups at the University of California, Berkeley's Haas School of Business and a research affiliate at the MIT Initiative on the Digital Economy.

Ramesh Johari is a professor of management science and engineering at Stanford University and an associate director at Stanford Data Science.

Sven Schmit is the head of statistics engineering at Eppo, an experimentation platform vendor.

Martin Tingley is the head of the analysis team on the Netflix experimentation platform.

Read more on **Experimentation** or related topics **Open innovation**, **Technology and analytics**, **Information management**, **Analytics and data science**, **Organizational change**, **Performance indicators** and **AI and machine learning**