



CITO RESEARCH

# Why Full-Stack Experimentation Is Essential For Digital Transformation

SPONSORED BY





# CONTENTS

<u>The Current State of Experimentation</u>	<b>2</b>
<i><u>Breaking through the limits of A/B testing</u></i>	4
<i><u>The difference between A/B testing and full-stack experimentation</u></i>	5
<u>Benefits of Full-stack Experimentation</u>	<b>7</b>
<u>Split's Role in Making Full-stack Experimentation Possible</u>	<b>9</b>
<u>Conclusion</u>	<b>10</b>



It is essentially mandatory for businesses to have an online presence. But AI pioneer [Andrew Ng](#) recently offered an important distinction: an online shopping mall is not an Internet company. To be a true Internet company, you have to be focused on improving your business and operations through the use of data and innovation, not just a storefront where someone can click on a product and buy something.

Full-stack experimentation is a core component of what it means to be an Internet company. To pinpoint what true full-stack experimentation is, it's helpful to illustrate what it is not. In the world of ecommerce, experimentation is commonly carried out using a set of practices that have come to be known as A/B testing. But full-stack experimentation and A/B testing are not synonymous. A/B testing is often done at a surface level, with a marketing focus on the user interface (UI) or with siloed offers.

**But full-stack experimentation has a much wider and deeper scope and is essentially the completion of a journey that started with agile development and has grown through DevOps, continuous integration, and continuous deployment.**

Full-stack experimentation serves not just marketers but developers and product managers seeking deep insights by comparing fully realized alternatives.

Full-stack experimentation adds missing elements to agile, DevOps, and CI/CD to allow those practices to move from the world of development to the world of data-based product development. It is not only about evolving software faster by putting up new features; rather, full-stack experimentation seeks to fully understand the impact of those features. A fast cycle time is great, but it does not provide the visibility needed in an experiment to assess causality and the *why* behind an outcome.

The missing elements needed to round out a complete approach to experimentation include:

- Support to change and run experiments on substantial elements of the application — not just the UI, but also code that implements features and controls process flows.
- Support for experiments that change code in services implemented in different languages and distributed across containers or other methods of distributed computing.
- Support for collecting and analyzing data from all available sources so that the comprehensive impact of changes and individual features can be measured and incorporated into product and operational metrics.



This practice is full-stack experimentation and it is widespread among leading Internet companies such as Uber, Amazon, Facebook, LinkedIn, Google, and Twitter, who have supported it using custom-built technology stacks. All of these companies constantly run experiments at all levels of their services, testing out better search algorithms, offers, and new processes.

These experimentation pioneers have the engineering and fiscal resources to create stacks and development tools to support multivariate experimentation. Smaller businesses and larger enterprises across verticals that don't have these resources, have, until now, not been able to pursue this type of experimentation. Yet rather than living in the A/B testing world of limited insights and assumptions about causes, all companies should be employing full-stack experimentation.

## What Is Full-stack Experimentation?

**Full-stack experimentation does not optimize part of a product or feature; it is about supporting experimentation as a new product development process. It is about allowing development to more directly support and inform the evolution of the business. In this way, full-stack experimentation formalizes and adds a mature, data-based method to the rapid evolution of agile, DevOps, and CI/CD.**

This white paper examines how businesses can embrace a culture of experimentation and make it part of their product development DNA. The paper will address the following questions:

- ① Why is full-stack experimentation crucial to sustaining effective innovation?
- ② What benefits come from full-stack experimentation?
- ③ How can full-stack experimentation be adopted broadly in an affordable manner?

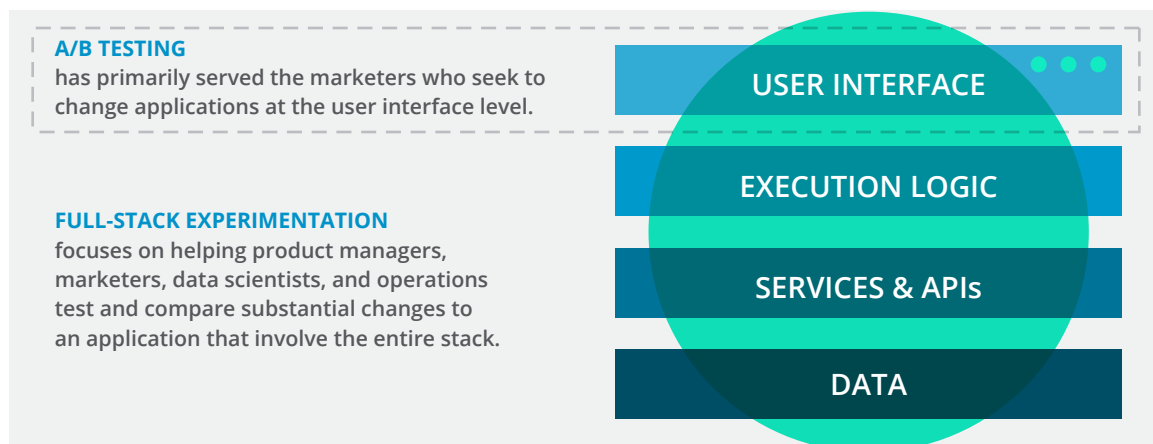
## The Current State of Experimentation

As any business that goes to market quickly learns, the way a product or feature is actually used can be radically different from the developer's intent. Customers respond to and use a product in the way that works best for them. The key challenge is for companies to learn what features or parts of a product appeal to their customers.



Experimentation in some form or another is not unique to the Internet. For instance, grocery stores and clothing retailers have always relied on testing, evaluating whether items sold better if they were located in one specific part of the store versus another. Movie studios and advertisers use focus groups. But the Internet has given companies the ability to vastly expand, improve, and revolutionize the experimentation process. In fact, you could argue that the core innovation of the Internet has been to take data generated by users at scale to perfect, personalize, and optimize the customer experience. And companies can realize true gains from many forms of experimentation. A [recent study by Qubit](#), the results of which were audited by Price Waterhouse Cooper, found that targeted, segmented offers can lead to a three-fold uplift for businesses.

Yet using some form of experimentation and actually implementing experimentation so that it drives tangible decisions about development and marketing are radically different. By adopting experimentation-driven product development, product development shifts from up-front guessing to a market-driven approach. At the same time, having the ability to constantly measure the impact of features on key metrics, update products based on that feedback, and improve the user experience is fundamental to success. Unfortunately, right now, most software delivery and outcomes measurement is done in departmental siloes within businesses. Instead, companies crave a type of experimentation that offers maximum transparency into how features are being used and the impact they have on the customer experience and on business metrics.



● = experimentation

**Note:** Some companies that practice full-stack experimentation refer to their efforts as A/B testing despite its larger scope.





## Breaking through the limits of A/B testing

For all but the largest business, the current state of experimentation of this kind is restricted and far from ideal. Companies widely use A/B testing, but it can be a very constrained version of experimentation. The limitations of A/B testing as it is commonly practiced are numerous:

- **A/B testing has a marketing orientation and is typically conducted at a surface level.** This was done because it became easy with JavaScript to conduct A/B testing on the UI. It appealed to marketers looking for ways to improve the user experience and drive increased engagement.
- **A/B testing doesn't typically introduce deeper changes** because it is designed to work on completed applications, often on UI elements. Companies may receive some feedback on a UI change, but they're unable to test at a deeper level — for instance, swapping out code to roll out a new feature to a small set of users and measure the results of that.
- **A/B testing can create an unwarranted sense of certitude.** Because A/B testing is often narrow in its scope, some lift can be achieved without companies recognizing the tradeoffs it can involve, like a hit to site performance that would cause the negative impact of the change to outweigh the positive.

A/B testing has its place, and it should be noted that some of the leading A/B testing vendors are aware of the limitations of current practices and are making efforts to address them. It's not that traditional A/B testing doesn't work, but rather that it offers a glimpse of a complicated reality. It's like taking a picture with a periscope — what is outside the frame far exceeds what can fit in it. A/B testing should generate an appetite for testing at deeper, more substantial levels.

To date, off-the-shelf solutions have mainly reinforced the marketing perspective of A/B testing. Full-stack experimentation has not been possible without the creation of a custom stack. Most existing solutions fall short in meeting the needs of a full-stack experimentation team. Such solutions simply don't allow engineers, data scientists, and product managers to experiment as much as marketers.

But that is now changing. Full-stack experimentation is now possible and differs greatly from traditional A/B testing.



## The difference between A/B testing and full-stack experimentation

Full-stack experimentation differs from traditional A/B testing in scope, purpose, and the potential impact on a business. Full-stack experimentation is focused on all those involved in the development of a product or application, from engineers to data scientists, as opposed to primarily marketers. It moves beyond the UI layer used in most A/B testing. You can use full-stack experimentation to understand what happens when you want to test deeper changes, like localizing or tailoring search results, switching credit card processors, or sending a user to an entirely new account creation flow.

Another key differentiator between full-stack experimentation and A/B testing is the number of variables that can be examined. Full-stack experimentation lets you test simple ideas as well as those that can have a large impact because they represent more substantial changes.

A former CEO of a leading ecommerce company that built its own full-stack experimentation platform summed it up this way:

Full-stack experimentation lets you test simple ideas as well as those that can have a large impact because they represent more substantial changes

*“Full-stack experimentation makes it easy to test small things and potentially find big wins without taking big risks. It’s a timesaver. You can throw out the bad ideas and can act on the good ideas quickly. Google and Facebook aren’t great companies because they always have perfect ideas. They’re great companies because they can test ideas quickly and throw out the bad ones while doubling-down on the good ones.”*

With full-stack experimentation, companies can always ask “What more can be tested and discovered?” The large scope of possible experiments unleashes creative thinking that goes beyond the UI.

The expanded range of data that supports full-stack experimentation is also a huge benefit. Remember, full-stack experimentation builds on agile/DevOps/CI/CD practices which include instrumenting a site so that when changes are made, any unintended consequences are quickly discovered. Full-stack experimentation builds on this approach and provides a wide range of instrumentation for experiments for the same reason.

**Take this example:** What if a company made a change with the intention of increasing its checkout flow, but in the process increased its site load time? With A/B testing, all the company would see is that they achieved their goal of increasing checkouts — they’d fail to realize that site performance degraded in the process. Full-stack experimentation offers a more comprehensive view of all the variables that affect any change or feature.



Full-stack experimentation isn't a panacea, but it does create a philosophical change in organizations that utilize it. Companies become more intellectually honest because there's greater transparency about what works and what doesn't. More importantly, full-stack experimentation is not about optimizing part of the product or features, but rather about supporting experimentation as a new product development process in which constant iteration and attempts at improvement become the guiding principle. This drives a culture of innovation. Full-stack experimentation allows development to directly support and inform the evolution of the business overall.

## Full-stack experimentation is the logical next step in the evolution of product development

Full-stack experimentation is not just a box for companies to check and be done with. It's a fundamental shift in how companies run their DevOps. Full-stack experimentation means that companies are using continuous delivery practices in which products and features are constantly tested and improved. This evolves the deployment process to a flow of small steps. How does this work?

Full-stack experimentation is the logical next step in an evolution that connects agile development and CI-CD, to the initial progress in experimentation made by possible A/B and A/B/n testing (which supports more than two alternatives). A true full-stack experimentation solution should offer rich aggregation and measurement metrics leading to clearer outcomes. Full-stack experimentation allows the same maturity in terms of ease of use and statistical evaluation of results from A/B/n testing. But the experimentation allows you to measure outcomes and determine correlation and causality in a way that is richer and more powerful, and not just at the surface or only in some aspects of the product.

The continuous delivery of full-stack experimentation offers numerous benefits. According to a [report by Zend](#), companies that use continuous delivery experienced a:

**21%**  
increase in new software and services delivered

**22%**  
improvement in quality of deployed applications

**19%**  
increase in revenue

**50%**  
fewer failures





## Benefits of Full-stack Experimentation

Full-stack experimentation allows you to build and bolster web applications using data science and analytics. Gone are the days of guessing whether a change caused an increase in sales. Full-stack experimentation allows you to answer these questions with rigorous results with statistical significance and high confidence rates.

Now that companies do not have to be an Internet behemoth like Google or Facebook to institute full-stack experimentation, they should consider the full main benefits of using this method:

**Create better products and instill confidence:** Companies need more experimentation to measure the impact of features on the customer experience in real-time. Customers will quickly abandon products that are not meeting their needs. Experimentation leads to better outcomes that are backed by data. You're removing the guesswork and can employ much more targeted sampling of groups and subgroups in your testing. This allows you to get to statistically significant results and avoid false positives. You are not just testing A versus B, but rather dozens or even hundreds of variables.

**Speed innovation and time to value:** Full-stack experimentation allows companies to constantly run small, targeted experiments. These experiments result in a swifter pace of releasing improvements. More people can be involved in designing and performing experiments that help to move a concept from an idea to an actual product. For instance, a [Qubit study](#) found that changes to sites that drive purchase impetus increased sales uplift by a factor of two to fourteen times over superficial UI alterations.

**The mitigation of risk:** Full-stack experimentation and continuous delivery mean testing can occur in a controlled environment with high degrees of segmentation. Changes can be rolled out to small subgroups of users, such as 1% of all users and improved based on feedback, before being gradually rolled out to 5%, 10%, and 50% of users. By the time the change is rolled out to 100% of users, all operational issues have been addressed.

**Better security:** Just as a constant iteration feedback loop of continuous deployment helps companies to avoid operational and performance risks, it also alleviates security threats. Frequent experimentation is a huge benefit to security because companies can move quickly to mitigate any problems that arise and address them immediately, before a product or feature has been deployed to all customers.



**Performance upgrades:** According to numerous CEOs and studies consulted for this report, in e-commerce, the main factors that affect conversion are the performance and speed of a site. With continuous delivery and full-stack experimentation, the larger range of metrics always includes those that track how the change will affect the performance of the site to avoid degrading the overall user experience and hurting the company's bottom line.

## Victories in full-stack experimentation

To date, full-stack experimentation has been used by industry leaders with the engineering resources to build their own framework for doing so. But the victories are substantial.

- Etsy has used continuous delivery for years as part of their efforts to consistently improve the customer experience on their site. The company continually runs experiments to assess what works and what doesn't — by one count, they run **50 tests a day**.
- Microsoft has **long relied on experimentation to fuel innovation**. They have used a model that takes a department from crawl to walk to run to fly stages with continuous deployment and testing to ensure a team fully embraces experimentation and can get the most out of it.
- Netflix **tests everything** from multiple angles. The Netflix Experimentation Platform is purpose-built for full-stack experimentation and is used by everyone from UI designers to developers to data scientists (interestingly, the company still calls this A/B testing).



## Split's Role in Making Full-stack Experimentation Possible

Split Software makes full-stack experimentation available to businesses of all sizes. Companies no longer have to create an in-house, custom built solution to experience the benefits of full-stack experimentation. Split offers experimentation for engineers, product managers, and data scientists.

Companies no longer have to create an in-house, custom built solution to experience the benefits of full-stack experimentation.

**How does it work?** Split's Feature Experimentation Platform offers the type of agile development and experimentation capabilities needed for today's enterprise technical teams.

Split unifies DevOps and product management practices, helping engineering and product teams accelerate the pace of product delivery and make data-driven decisions through its robust feature flagging and extensive experimentation capabilities.

Split is also the only platform that can do continuous delivery and full-stack experimentation in a single unified platform.

A crucial component of the Split platform is that it allows companies to have confidence in the results of experiments. Split offers detailed feedback on experimentation: it's not just an experimentation platform, but also a results engine, which allows companies to understand what is truly working and what is not. With Split, data about behavior can be collected, consolidated, and automatically analyzed across experiments. Split therefore becomes the system of record for all development, product, and business metrics within an organization, unifying the experimentation process.

Split offers a new lens on BI and business analytics. With the feedback the platform provides on the results of experimentation, companies have a new way of understanding the impact feature flags or product changes have on certain metrics and the ROI of certain functions within their existing investments.

Split also enables companies to use trunk development. This means that companies can take code that is still in the process of being committed and finalized, and share it behind a feature flag that's turned off, which prevents new code from breaking the test. Companies have the ability to constantly iterate on what variation of an experiment they want to show customers, even if the variation is not fully completed.



Additionally, Split enables an environment of continuous feature development and experimentation. This type of experimentation helps create more effective applications because companies learn how users are interacting with their product and are then able to iterate on that knowledge.

## Conclusion

For too long, full-stack experimentation has been the province only of companies with deep pocketbooks and even deeper engineering teams. Now full-stack experimentation, and the continuous delivery and feature flagging it enables, is available to companies of all sizes. Companies can now expand their testing beyond the realm of front-end marketing. They can run constant experiments to see what will help to optimize performance of their site and their user experience, and do so in a controlled way that ensures problems are addressed before they are made available to all users. And companies can have confidence in the statistical rigour of their experiments, quantifying the results of changes in a way that A/B testing cannot match.

Companies can now expand their testing beyond the realm of front-end marketing.

Try [Split for free](#) or [contact them](#) for a personalized demo.

This paper was created by CITO Research and sponsored by Split

### CITO Research

CITO Research is a source of news, analysis, research and knowledge for CIOs, CTOs and other IT and business professionals. CITO Research engages in a dialogue with its audience to capture technology trends that are harvested, analyzed and communicated in a sophisticated way to help practitioners solve difficult business problems.

Visit us at <http://www.citoresearch.com>