

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

by Christopher Condo and Bill Seguin

January 15, 2019

Why Read This Report

Most application development and delivery (AD&D) teams are stuck guessing about what their users need, relying on internal experts to guide their decisions. This approach to development is no longer sufficient to keep up with changing user preferences or increasingly complex production environments. Read this report to learn how experimentation platforms can shorten time-to-value by enabling dev teams to test ideas directly with customers, experiment with functional parameters, and release service components in a safe, controlled manner.

Key Takeaways

Without Experimentation, There Can Be No Innovation

Experimenting in production introduces many risks; that's why most teams rely on subject matter experts (SMEs) to guide their efforts. However, expert opinions get you only so far. Without the ability to try ideas with real users, there's no innovation, just slow refinement.

Experimenting In Production Speeds Innovation

Experimentation platforms provide a safe way to try new ideas in production, with real users and real data. This allows AD&D teams to push innovative ideas to end users faster, let users determine winning approaches, and settle internal arguments once and for all.

Experimenting In Production Is The New Normal

Getting ideas in front of users faster — “continuous discovery” — is the new normal. Experimentation platforms let teams test out minimum viable products (MVPs) in real-world scenarios with precise control.

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

by [Christopher Condo](#) and [Bill Seguin](#)

with [Christopher Mines](#), Andrew Dobak, and Andrew Reese

January 15, 2019

Table Of Contents

- 2 AD&D Teams That Aren't Experimenting Aren't Innovating
- 2 Experimenting In Production Drives Faster Innovation
 - Feature Flags Enable Continuous Delivery With Lower Risk
 - A/B Testing Lets Customers Choose The Superior Experience
 - Direct User Testing Gives Engineering Teams Early Clues For Winning Designs

Recommendations

- 10 Experimenting In Production Is The New Normal
- 11 Supplemental Material

Related Research Documents

- [Apply These 7 Steps To Get Software Release Velocity Unstuck](#)
- [Digital Transformation Requires Development Transformation](#)
- [Master Your SDLC For Modern Application Delivery](#)



Share reports with colleagues.
Enhance your membership with Research Share.

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

AD&D Teams That Aren't Experimenting Aren't Innovating

Most software teams are never really sure if they are making a difference to their users, except when the customer support line lights up and they realize that difference was a negative one.¹ Dev teams often operate in a fog, taking best guesses at which path to take to improve their applications (and their business). But now that software-based experiences have become differentiators for every business, the pressure increases on AD&D teams to produce winning results faster — with fewer mistakes.² Experimentation platforms provide the insight dev leaders need. Unfortunately, most organizations don't yet use them, and so they find:

- › **Replicating production environments is complicated.** Modern applications are complex, composed of monoliths, microservices, cloud, and private data center components. In this tangled app ecosystem, there's more to changes than the size of new features. A Honeywell AD&D leader told Forrester that to best forecast the impacts of changes, the organization “needed a way to safely test high-impact changes in production.” Teams often use service virtualization to mimic production, a step no one should skip. But for highly complex or risky changes, teams must also be able to safely test in production while still minimizing risk to business and customers. As Eric Cane at Envoy told us: “Recognize that the only source of truth is production.”
- › **No news is bad news.** AD&D teams that operate under the mantra of “no news is good news” never know if they are creating a better experience for their end users — only that the sky didn't fall on the last release. Most teams rely on business analysts to let them know, months later, if they moved the needle, but by that time they've missed out on critical opportunities to learn.
- › **AD&D schedules determine release dates, not the business calendar.** Typical feature releases to deployment are filled with stress, anxiety, and frustration. The business wants to coordinate marketing and communications while battling development teams to commit to a release date. After long hours and late nights, companies follow the “press and pray” mantra of release, desperately hoping no bugs arise in production. This type of pressure doesn't help either part of the organization, and it leads to development teams making hasty decisions to hit a release date and marketing pros to hold back on feature release promotions. As Timothy Chabot at BMW described the situation, he “needed a way to allow the marketing team to control the launch of new features.”

Experimenting In Production Drives Faster Innovation

Without a safe way to experiment in production, AD&D teams take the safest path, but that isn't necessarily the best path to drive more business. Experimentation platforms mitigate the risk of trying new ideas in production by providing precise control of feature release, feature comparison, and functional comparison — while in production with actual users. With this peace of mind, teams are free to try new ideas and experiment with more innovative and potentially disruptive solutions.

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

AD&D pros have several options when it comes to picking a style of experimentation platform. They can choose between feature flagging, A/B testing, and direct user testing (DUT) tools (see Figure 1). Each tool offers a different flavor of experimentation, and while one type of tool may provide value by itself, top performers use them all to maximize the impact of their efforts.

These forms of experimentation can advance the state of engineering and application design for any AD&D team.

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

FIGURE 1 Three Kinds Of Experimentation Platforms AD&D Pros Can Leverage To Gain Insight Directly From Users

	Feature flagging	A/B testing	Direct user testing
Characteristics	Allows application code to be released in “off” mode and turned on later via dashboard.	Enables in-production testing of different versions of an application. Determines winning strategies within hours or days.	Real people directly engaged and encouraged to speak their thoughts out loud as they navigate workflows, try new features, or look at new UX.
Purpose or goals	Enables single-trunk development; features are continuously deployed but only turned on when ready; shown to a select audience based on data.	Compare new work flow patterns, search engines, algorithms against the current version to see if improvements can be made.	Understanding the “why.” A/B testing can tell which features win, but they can’t tell you why. DUT does that.
Benefits	Simplifies release, enables continuous delivery, enables business team to manage the release date, enables user demographic splits.	Experimenting with different patterns is simplified and organic part of development. Statistical engines can immediately inform devs which strategy is performing better.	Glean insight directly from people and overhear their assumptions, allowing teams to better meet or exceed expectations.
Risks or downsides	Flags need to be maintained. Forward thinking needs to happen (when and where to place these flags) to get the most value, and re-engineering is needed to place them into legacy code.	Switching logic needs to be maintained and removed after experiments are completed. Loosing code need to be removed from the code base.	You need to recruit and reward people for their time. Finding suitable test subjects for testing skill-based applications (like a CAD tool) might be difficult.
Sample vendors	LaunchDarkly, Split, FF4J, Toggly	AB Tasty, Google Optimize, Optimizely, Adobe Target, VWO, Oracle Maxymiser	UserTesting, Userlytics, usertest.io

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

Feature Flags Enable Continuous Delivery With Lower Risk

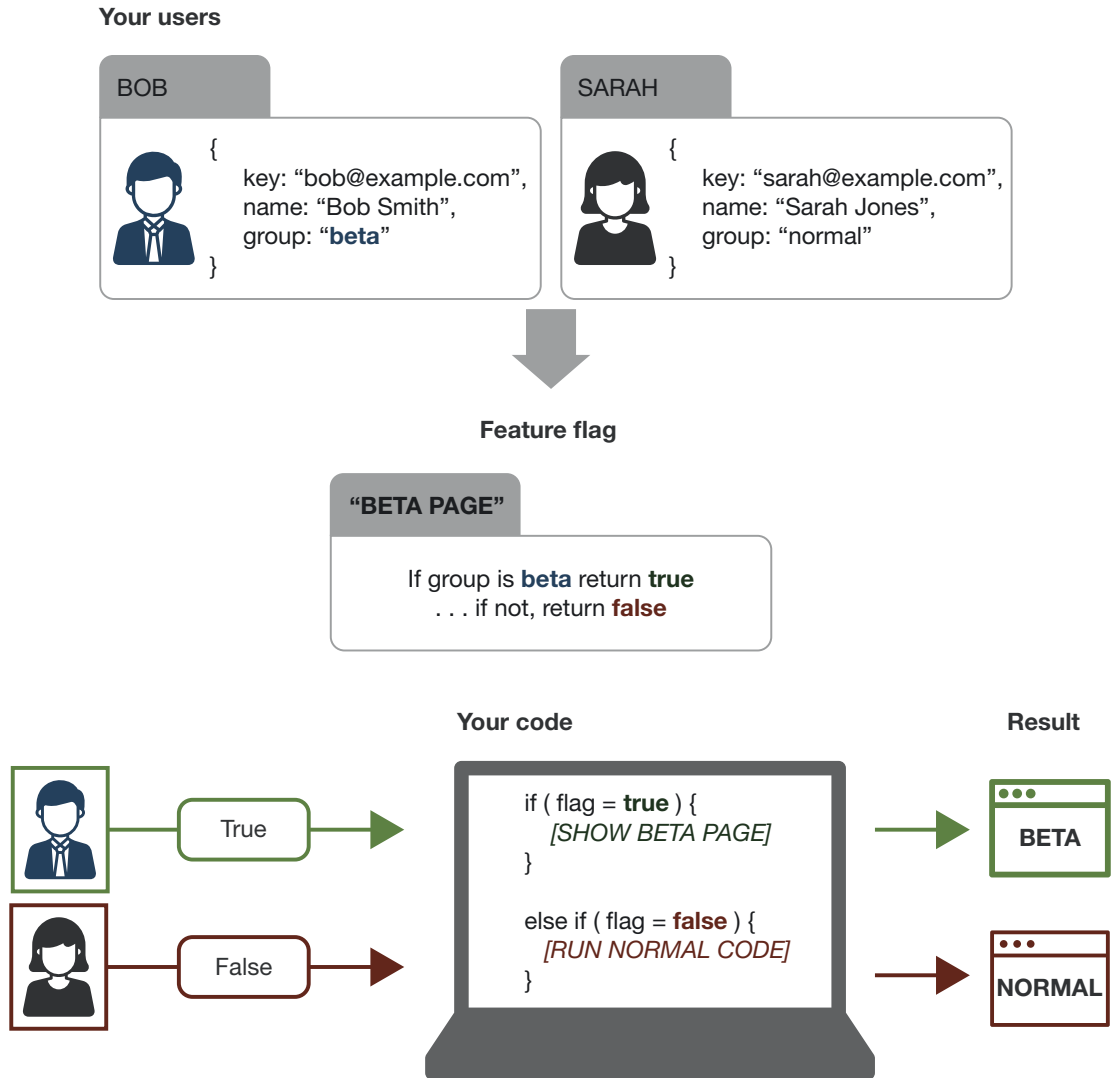
Feature flags let AD&D teams to develop code as they normally would, allowing them to version control, test, and deploy the same as always. However, feature flags give those teams complete control of when to expose a particular feature to users, allowing administrators to turn it on and off at the flip of a switch (see Figure 2). Feature-flag platforms consist of the following components: if-then-else logic embedded directly into application code to switch features on or off in production; an application or service that communicates the value of the feature flag to the application; an administrative console for managing all the feature flags; and the ability to configure which user populations see each feature when it's flipped on.

AD&D pros tell us that that feature flags have made it easier for them to manage feature development by allowing them to:

- › **Decouple feature release from feature deployment.** Developers deliver the code, but business teams release it. Feature flags allow business leaders to govern which features are exposed, when they expose them to end users, and what customers they expose them to. This is key to creating differentiating experiences for customers. One dev team we spoke with leveraged this ability to manage variations of a product via configuration variables and multivariate feature flags based on users' locations.
- › **Enable single-trunk development.** Single-trunk development is growing in popularity. This methodology merges all code into a main trunk as early as possible in the AD&D cycle, eliminating costly merges that typically happen late in the cycle. This has been particularly effective for a financial services customer we spoke with; using Split, it can “work much faster [because the team] doesn't need to maintain as many branches or merge pieces.”
- › **Improve risk mitigation.** Feature flags help teams follow progressive release strategies. AD&D teams can fully control every feature flag from a central dashboard and can even implement rule-driven automation to shut down errant code before problems spiral into a support incident. Using LaunchDarkly feature flags, a DevOps lead at Honeywell could “manage large features and high-impact changes, like changing entire data flows or migrating to a new database type.” He could improve mission-critical technologies significantly and attests: “We would not have been able to do it ourselves.”

FIGURE 2 Feature Flags Lower Risk

Displaying a beta page to users



AD&D Pros: Stop Guessing And Start Experimenting

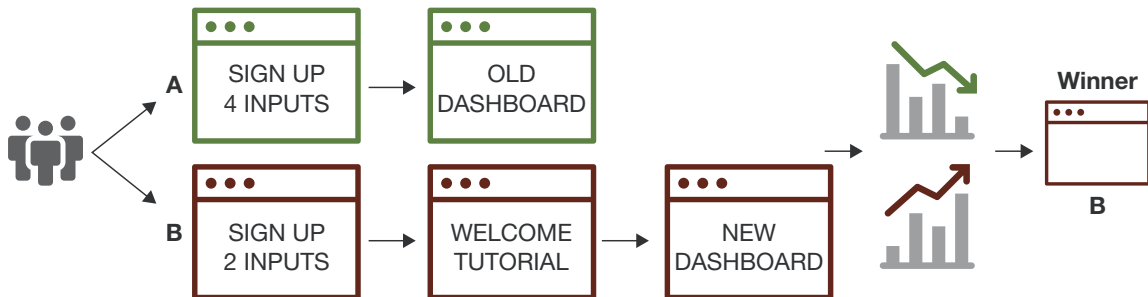
Continuous Improvement: The Modern Application Delivery Playbook

A/B Testing Lets Customers Choose The Superior Experience

A/B testing is a well-known tool for marketing and UX design teams. Experience designers already use it to make graphical changes to app experiences and measure their impact. But A/B testing platforms have evolved to become more relevant for AD&D teams and software developers. Platforms can now enable direct, comparative testing on the server side that goes deeper than the familiar client-side testing of UI content (see Figure 3). A/B testing platforms consist of the following components: A/B testing logic embedded in code; a system that controls which populations see experience A versus experience B; and a statistical engine for determining the winning experience.

A/B testing platforms can determine winning tests in a matter of hours or days, allowing teams to quickly ditch bad ideas in favor of better ones. This deep comparative testing allows AD&D pros to:

- › **Execute tests by comparing new and old features with similar user populations.** Valid A/B comparisons require testing with similar populations to make the results meaningful. A/B testing platforms provide controls to scope the population and manage the experiment to ensure the results have statistical significance and provide insight for making improvements. This doesn't mean experimenting sitewide; one Optimizely customer ran reliable A/B tests by splitting a group of users that made up about 10% of its total traffic.
- › **Determine optimal user workflows.** Working with an experimentation pro, AD&D teams use A/B testing to compare new workflows against existing ones to continuously improve UX and extract the greatest business value from users. A/B platforms allow them to adjust the mix of old and new to limit potential negative exposure; they also allow them to expand as needed if results look good, all from a console rather than costly deployments and redeployments. This was especially helpful for a financial services client whose website changed daily to account for interest and loan amount variations. With Optimizely, it could make those changes and track and measure their impacts directly.
- › **Compare functional parameters in production.** With A/B testing, AD&D pros can set parameters for complex algorithms so it can test different, randomized ones side by side in production under similar conditions (such as traffic and load), helping engineers discover the best parameters. Such a task would traditionally require deployment, monitoring, and analysis of telemetry data. A/B undercuts that rigamarole, providing answers more quickly and easily. Mike Dikan, software engineering manager with edX, provided an example: "We use Optimizely for backend server optimization, for example to trigger and test between substantial changes in the user experience of our application."

FIGURE 3 A/B Tests Allow Two Different Experiences To Be Tested At The Same Time**A/B testing — compare new features, experiences, and application functionality****Direct User Testing Gives Engineering Teams Early Clues For Winning Designs**

DUT was once only employed by marketing teams (think Coke-versus-Pepsi blind taste tests). It involves real people, sitting down in labs or on their computers, testing out new, mocked-up ideas while they are observed by product designers (see Figure 4). DUT has the following components: real people invited to participate in testing; a lab or platform that allows user to be recorded as they test your application; and a well-thought-out experiment or hypothesis for users to test.

Multidiscipline product teams looking to gain insight and test new ideas earlier and less expensively should consider “shifting right,” bringing this experimentation methodology into their product development process earlier on. That means collaborating with experimentation pros and experience design pros to:

- › **Learn before they code.** Simply put, DUT saves companies times and money by forcing them to focus on building the right software. Product development pros use a range of low-cost mockup tools, from pen and paper to wire frames, to try out new ideas on users before code is even written. At Harry’s Razors, “[We] simply wrote all the product names on 3-by-5 index cards and had users place them in piles of like products. That helped [them] understand how users think about [their] products and how to structure menus so they made more sense to [their] users.”
- › **Observe users navigating mocked-up applications.** Ideas that have progressed to this stage can be mocked up with low-code or other UI tools to try out actual user flows, verbiage, and page designs. Again, this saves time and money by trying out patterns, designs, and other UI elements before the costly expense of actual development.
- › **Encourage test users to describe their thoughts out loud as they navigate the app.** Possibly the best reason to add DUT to your experimentation portfolio is that you can watch and listen to users as they navigate an application and explain out loud their thought process throughout the

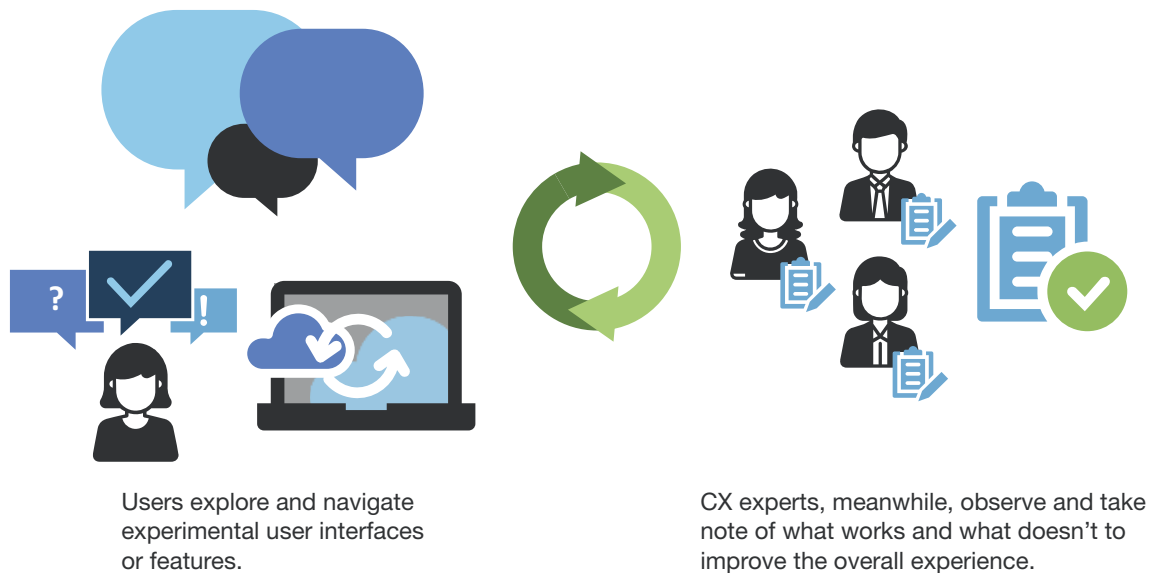
AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

entire exercise. This allows teams to learn why users make selections or run into trouble using a particular design. This uncovered hidden confusion for Harry's when it created an experiment for a "trial package" and discovered that end users always assumed that a "trial" implied "free" (which it wasn't). Because of this direct user feedback, Harry's could change the wording to better align with expectations, renaming it "starter package" instead.

FIGURE 4 DUT Tools Provide Clarity To Developers

Direct user testing lets customer experience (CX) experts observe and listen to users to determine optimal experiences.



Recommendations

Experimenting In Production Is The New Normal

AD&D leaders need to give thoughtful consideration to moving forward with experimenting in production. Simply running random experiments is wasteful at best and potentially brand-damaging. Further, managing large infrastructure changeovers will require engineering effort to imbed feature flags up and down your tech stack. So to get started with experimentation platforms:

- › **Create a premise to narrow down choices.** Experts tell us that just starting to throw ideas onto production servers to see which ones stick is a big mistake. Instead, work with an experimentation professional to formulate a premise about how and why a service, flow, or algorithm needs to be improved, create an alternative, and test it against the baseline. Using an A/B tool, adjust the percentage of population that sees the alternative experience. Usually, you'll be able to identify a winning strategy in hours or days.
- › **Get the pros involved for high-impact changes.** Consult with senior tech staff when considering high-impact modifications to applications. These types of modifications are usually complex, such as updating service layer logic or trying out a new database schema, and will require planning and surgical changes to each layer of the application stack in order to isolate the test version of the logic from the baseline.
- › **Know that building your own experimentation platform can quickly become impractical.** Experimentation platform vendors tell us their best customers are often those that tried creating their own testing platform but came to realize their one-off idea had become a product unto itself, requiring its own team and budget to further develop and maintain functionality. For these teams, the build versus-buy-decision turned in favor of buy after seeing the complete value vendor offerings provided.
- › **Don't use experimentation as a substitute for tried-and-true testing.** The move to experimentation does not mean that you can ship a feature into production with no oversight. Testing will continue to be a crucial focus for AD&D orgs despite the added benefit of being able to extend that testing out to production.³ Additionally, some system changes may be too complex to manage with feature flags or too expensive to break apart. If that's the case, take a look at application release automation vendors with blue/green testing automation that allow you to control the user population and provide automated rollback.
- › **Understand that experimentation creates technical debt.** Once your experiment is complete, it's important to clean up the switching logic, remove unused code paths, and keep everything tidy. Not doing so creates technical debt in the form of unmaintained and unused code. It's not just sitting there, it's using up CPU cycles on wasted build and test automation runs — worse yet, it could become an attack vector for hackers poking around for vulnerabilities.

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

Engage With An Analyst

Gain greater confidence in your decisions by working with Forrester thought leaders to apply our research to your specific business and technology initiatives.

Analyst Inquiry

To help you put research into practice, connect with an analyst to discuss your questions in a 30-minute phone session — or opt for a response via email.

[Learn more.](#)

Analyst Advisory

Translate research into action by working with an analyst on a specific engagement in the form of custom strategy sessions, workshops, or speeches.

[Learn more.](#)

Webinar

Join our online sessions on the latest research affecting your business. Each call includes analyst Q&A and slides and is available on-demand.

[Learn more.](#)



Forrester's research apps for iOS and Android.

Stay ahead of your competition no matter where you are.

Supplemental Material

Companies Interviewed For This Report

We would like to thank the individuals from the following companies who generously gave their time during the research for this report.

AB Tasty

LaunchDarkly

Adobe

Optimizely

BMW

Oracle

Envoy

Split

Google

UserTesting

Harry's

Wingify

Honeywell

AD&D Pros: Stop Guessing And Start Experimenting

Continuous Improvement: The Modern Application Delivery Playbook

Endnotes

- ¹ When we asked global developers how they measure their success, 35% said they use direct measurement of user/customer satisfaction; 38% use indirect measurements like number of defects fixed, defect density, and test coverage. Source: Forrester Analytics Global Business Technographics® Developer Survey, 2018.
- ² See the Forrester report “[Make The Case For Agile And DevOps-Driven Digital Transformation.](#)”
- ³ See the Forrester report “[How Developers Deploy Test Automation To Enable Continuous Software Delivery.](#)”

We work with business and technology leaders to develop customer-obsessed strategies that drive growth.

PRODUCTS AND SERVICES

- › Core research and tools
- › Data and analytics
- › Peer collaboration
- › Analyst engagement
- › Consulting
- › Events

Forrester's research and insights are tailored to your role and critical business initiatives.

ROLES WE SERVE

Marketing & Strategy Professionals

CMO
B2B Marketing
B2C Marketing
Customer Experience
Customer Insights
eBusiness & Channel Strategy

Technology Management Professionals

CIO
› Application Development & Delivery
Enterprise Architecture
Infrastructure & Operations
Security & Risk
Sourcing & Vendor Management

Technology Industry Professionals

Analyst Relations

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866-367-7378, +1 617-613-5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.