



Ebook



An Intro to Product Experimentation and Feature Management

Contents

1. How Continuous Development Turns Product Managers into Experimenters

- Move Fast, Risk Less
- Your Customers, Not Your HIPPOs, Decide
- Get off the Ford Line
- Marketing and Product Manager Alignment
- Developers Focus on Core Business Objectives

2. What's in It for Your Marketing, Product and DevOps Teams?

3. The Signs Your Company Needs Product Experimentation and Feature Management

4. Common Applications for Feature Flagging That Will Make Your Life Easier

- A/B Test
- Personalization
- Feature Flagging/Toggling For Marketing Alignment
- Progressive Rollout

5. Building vs Buying, and How to Choose a Solution

AB Tasty

The customer experience optimization company



How Continuous Development Turns Product Managers into Experimenters

```

8     e.preventDefault();
9 }
10
11 //on pressing return, addImage() will be called.
12 $("#urlBox").submit(addImage);
13
14
15 // editing image via css properties-
16 function editImage() {
17     var gs = $("#gs").val(); // grayscale-
18     var blur = $("#blur").val(); // blur-
19     var br = $("#br").val(); // brightness-
20     var ct = $("#ct").val(); // contrast-
21     var huer = $("#huer").val(); //hue-rotate-
22     var opacity = $("#opacity").val(); //opacity-
23     var invert = $("#invert").val(); //invert-
24     var saturate = $("#saturate").val(); //saturate-
25     var sepia = $("#sepia").val(); //sepia-
26
27     $("#imageContainer img").css("filter", 'grayscale(' + gs+
28                                     '%) blur(' + blur +
29                                     'px) brightness(' + br +
30                                     '%) contrast(' + ct +
31                                     '%) hue-rotate(' + huer +
32                                     'deg) opacity(' + opacity +
33                                     '%) invert(' + invert +
34                                     '%) saturate(' + saturate +
35                                     '%) sepia(' + sepia +
36                                     '%)');
37
38     $("#imageContainer img").css("-webkit-filter", 'grayscale(' + gs+
39                                     '%) blur(' + blur +
40                                     'px) brightness(' + br +
41                                     '%) contrast(' + ct +
42                                     '%) hue-rotate(' + huer +
43                                     'deg) opacity(' + opacity +
44                                     '%) invert(' + invert +
45                                     '%) saturate(' + saturate +
46                                     '%) sepia(' + sepia +
47                                     '%)');
48 }

```

20 years ago, tech companies were hit with the 'Agile Revolution'. The idea? Shipping working software every week or two would help teams deliver better products, even if this method implied more risk. In other words, the 'move fast and break things' mentality reigned.

But that was two decades ago. Today, agile is mainstream, and new philosophies, building on the agile movement, have come to the fore; namely, Continuous Integration, Delivery and Deployment, largely geared towards DevOps teams. Their big draw is that these processes and tools automate quality assessment, assuring that when code is merged in piecemeal fashion – and not on one big bang release day – it works. Even better, software can be deployed to the product environment at any time, by anyone. Now, your product manager can take the reins.

Today, the market is ready to go a step further. From Agile to Continuous Integration, Delivery and Deployment comes a thirst for Continuous Development. Continuous Development – we could even call it Continuous Activation – encompasses all of these ideas, but takes the logical next step. It puts even more control and autonomy in the hands of Product Managers. It allows them to not only deploy software themselves, (with mitigated risk), but also to pick and choose according to their own prerogatives which audiences are exposed to a given feature at any given time. In other words, they can run experiments, personalize the user experience, and exercise complete rollback control based on real-time data.

Continuous Development platforms and processes transform the Product Manager into a Chief Experimentation Officer, and there are many reasons to embrace this new paradigm shift:

AB Tasty

The customer experience optimization company



Move Fast, Risk Less

'Move fast and break things' only works if you're willing to accept the consequences of what you've broken. Most software developers would still like to move fast, but without the risk.

Continuous Development and the tools that support it factor in risk assessment. By avoiding code merges on one big release day, and by enabling progressive rollout techniques (canary testing, ring deployments), developers can avoid putting all of their metaphorical eggs in one basket. If your system has a feature flagging or rollback KPI embedded in the platform, switching off a defective or negative feature can be done instantaneously and painlessly.

AB Tasty

The customer experience optimization company

Your Customers, Not Your HIPPOs, Decide

How do decisions get made in your tech company? Chances are, HIPPOs, new bosses, vocal salespeople, consulting groups or the noisiest Product Manager in the room dominate that discussion, letting their personal experience, gut feeling or intuition determine the road map.

With Continuous Development platforms, the focus shifts from subjective ideas to customer feedback and data. Early adopter programs, beta testing, progressive deployment, A/B tests... all of these methods, enabled by feature flagging, toggling, and other Continuous Development techniques, make your main measurement of success the behavior and opinions of your customers.

In a B2B context, this might look like extensive interviews with early adopters. In B2C, it's likely your support teams or community manager who will pick up on positive or negative feedback around a new feature launch. Either way, Product Managers get direct access to the Voice of the Customer and can form data-driven arguments for why to rollback, stick with or modify a new feature.



AB Tasty

The customer experience optimization company



Get off the Ford Line

If your team is project-driven, chances are your Product Managers and developers feel they need to keep their heads down and noses to the grindstone, working on their piece of the software production puzzle. They might be productive, they might be agile, but they might also not really feel the business impact of what they're working on 40+ hours of the week.

When you can experiment with and test the features you're developing; when you can get direct user feedback and adjust your work accordingly; when you have clear, measurable KPIs that determine success, your work all of a sudden feels a lot more meaningful. This keeps teams motivated, fresh and loyal.

AB Tasty

The customer experience optimization company

Marketing and Product Manager Alignment

When you give your Product Managers more control, it's easier for them to align with the teams around them, especially the Marketing and Communication departments. A new feature release, especially depending on the size and importance of your company, can mean a big web of marketing and communications campaigns. Emailings, press releases, articles, social network posting, corporate website updates... retroplannings and shifting deadlines are much easier to manage when your Product Managers are in the driver's seat and not beholden to developer teams that have other priorities and are even more far removed from your marketing and communications personnel.



Developers Focus on Core Business Objectives

If you have a robust developer team, there's a chance you could set up these types of feature management systems in-house, without the need for a dedicated platform. But this is time-consuming, and one could argue that it diverts skills and resources away from your core business objectives.

I believe that the time is now for Continuous Development.

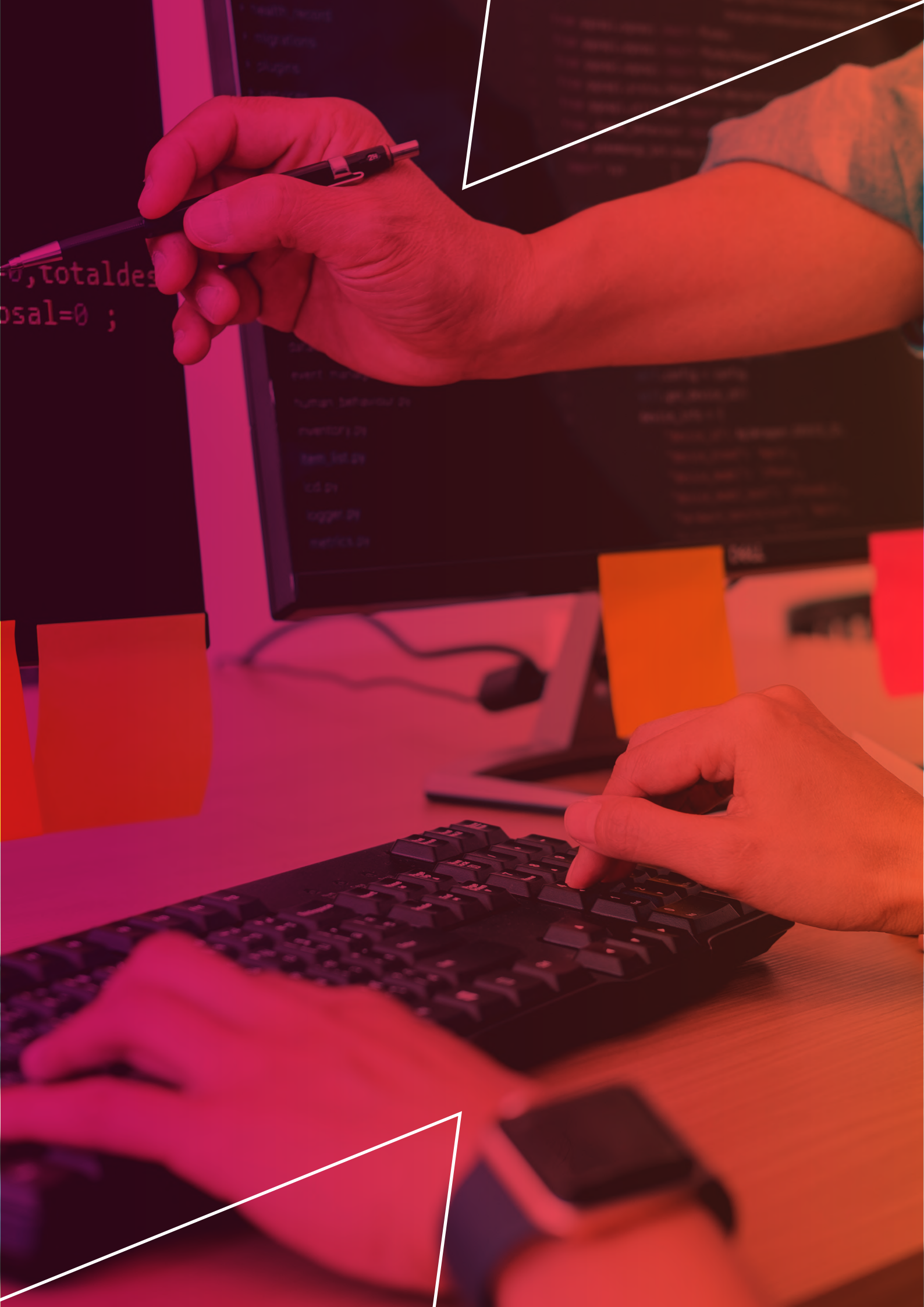
By turning our Product Managers into Experimenters, we're able to build a better product and bring it to market faster, with less risk; we continue in the vein of 'customer obsession'; we keep our teams creative and motivated; and we generally build up what, at AB Tasty, we've been advocating for since our founding – a test and learn, experimentation culture.



-Jean-Yves Simon, VP Product at AB Tasty

AB Tasty

The customer experience optimization company



What's in It for Your Marketing, Product and DevOps Teams?

Definition

Feature flags (also known as feature toggles) use conditional logic to let you modify system behavior without having to ship new code. Here is what a basic implementation looks like in your code base using Flagship, our product optimization platform:

```
// The flagshipData variable holds the API response with all the test and
// variation data associated with the userContext
const flagshipData = callFlagshipDecisionAPI(userContext);
// The betaChecker function is a toggle point that enables a new feature
// based on the value of the cohort flag in the Flagship API response
function betaChecker(flagshipData) {
  if(flagshipData.beta) {
    // Display new feature
  } else {
    // Display old feature
  }
}
```

Essentially, feature flags allow for better alignment between Product, DevOps, Developer and Marketing teams by *disassociating code delivery from feature release*.

AB Tasty

The customer experience optimization company

In addition to the fact that you can ship a totally new feature to your users, you can also flag variables to modify them after deployment. This is called a Remote or Dynamic variable, because you can set its value on the fly. Here's what implementation with Flagship looks like:

```
// The flagshipData variable holds the API response with all the test and  
variation data associated with the userContext  
const flagshipData = callFlagshipDecisionAPI(userContext);
```

```
// The flag variable is set dynamically to a variation returned by Flagship  
or a default value  
const flag = flagshipData.modificationKey || 'default';  
// Use those flags later on in your code  
document.getElementById('flagged-element').innerHTML = flag;
```

Long story short, feature management and product optimization is about increasing control, speed and precision, while decreasing risk, when it comes to software development.

It makes developer teams happy because they'll be solicited less frequently by Product Managers, who are given greater autonomy. Product Managers are happy because they can exercise more control when it comes to feature rollouts, as well as adopt the experimentation mindset that their marketing colleagues have been practicing for ages. And marketing teams are happy because they can align more easily with product teams when it comes to scheduling communications surrounding product launches or updates - ultimately allowing companies to deliver an even more well-rounded, optimized and unified customer experience.

Considering how marketing teams have long used client-side solutions to increase conversions - it would be a shame for the product to not live up to the bargain, as [Lindsay Kwan writes for WiderFunnel](#). "If you are optimizing your website to get a customer to convert, only to have a product that does not live up to the customer's expectations—those hard-to-win customers are going to look elsewhere."

AB Tasty

The customer experience optimization company

The Signs Your Company Needs Product Experimentation and Feature Management

- Delivery schedules slip from one week to two, three...your Product Team feels pressured to speed things up, while at the same time avoid any bugs or mistakes.
- You're not confident certain product features you're developing are really increasing business output or improving the customer experience.
- You've had problems in the past merging code branches, (you prefer a single trunk method), or wish to alleviate the risk involved in single release days.
- You make frequent updates that affect infrastructure, meaning they are high risk.
- Your product and developer teams complain about being kept to a marketing agenda or timetable; your marketing team complains about delays and lack of control or visibility.
- You want to expand your current marketing site experiments by running omnichannel campaigns and take those learnings to personalize experiences across every touchpoint.
- You've gone through at least one major, painful rollback.
- You lack data-driven reasons for developing or freezing development of certain product features.
- You keep hearing that the company wants to become more customer-focused.

AB Tasty

The customer experience optimization company

- Your DevOps or QA team finds it difficult to replicate production environments - they wish they could safely test in production on real users in real environments.
- Developers feel in the dark about whether their projects actually 'made a difference' to the customer experience or the business bottom line - feedback comes in too late to feel relevant or useful.
- The C-Suite wants to go faster and innovate more, without however wanting to 'break things' along the way.
- Your company is product-led (or headed in that direction) and you need a platform that allows business and technical stakeholders to collaborate.



AB Tasty

The customer experience optimization company

Common Applications for Feature Flagging That Will Make Your Life Easier



Feature Management and Product Experimentation platforms based on feature flagging techniques open up a variety of possibilities for Product Managers to increase speed, control and precision when it comes to product releases, while at the same time decreasing associated risk. Here are a few frequent scenarios:

A/B TEST

Let's say, as a Product Manager at a large e-commerce site, you're considering changing your pricing structure. You don't know which of two ideas will generate the most revenue, which is your main concern.

With a product experimentation solution in place, you can easily run an experiment in which you deploy your set of code with the new feature - wrapped, of course, in a feature flag.

AB Tasty

The customer experience optimization company



Running your experiment in the same way you would run a classic A/B test - randomly assigning similar and equal traffic to variation A and B - with a primary KPI (revenue) will allow you to assess objectively which pricing structure is better. The toggle will let you choose to run variation A with the feature 'off', and variation B with the feature 'on.'

In this scenario, you'll naturally want your audience traffic to be as similar as possible, though you may run the experiment on a low-risk or specific audience, and not your entire client base.

You'll notice that you as a Product Manager you don't need to directly solicit your DevOps to run this test, if you're using an efficient feature management platform. A centralized console will give you the autonomy to set up, choose targeting, run (toggle on and off) and assess the impact of your test, autonomously and without needing to ship new code.

In this way, Product Managers can run their own experimentation - and by virtue of that fact, optimization - cycle. They can gather their own test hypothesis based on their experience, customer feedback and cold, hard data. They can act on those ideas virtually independently thanks to a centralized console. And they can analyze these results with measurable primary and secondary KPIs.

In fact, any kind of experiment that can be done server-side - whether from a desire to ensure maximum performance and avoid even the tiniest chance of a flickering effect - can be done with a Product Experimentation solution. The difference is that using the latter greatly facilitates the process, especially if there are pre-defined templates and use cases set up in the platform.

AB Tasty

The customer experience optimization company

PERSONALIZATION

The same functionality that powers feature toggles can also power personalization scenarios. Developing personalized product experiences can require hundreds of lines of code to account for all possible targeting criteria and their corresponding variations. Using feature flags/toggles, only default scenarios need to be hard-coded — the targeting criteria and code variations can be configured remotely to be automatically applied at runtime. The result is cleaner code and finer control over the end-user experience.

As a Product Manager, do you have a feature you only intend to enable for 'Pro' users of your SaaS platform? Or a feature that only interests one of your three buyer personas? Or perhaps it's a feature that is only relevant for one geographic area.

Feature management tools will likewise enable you to target which audience can benefit from which features, again without resorting to new code.



FEATURE FLAGGING/ TOGGLING FOR MARKETING ALIGNMENT

Ever hear your developers complain about having to keep to a marketing agenda? Use feature flagging to turn 'on' a new feature to align with a press release or communications schedule, instead of keeping your developer team walking on eggshells to release in real time.

AB Tasty

The customer experience optimization company

PROGRESSIVE ROLLOUT

So, you've got new code you need to rollout. Feature Management lets you uncouple software delivery from feature release - and in so doing, mitigates the risk factor.

Turn 'on' your feature flags to a select and restricted audience - perhaps you'll decide it's most judicious to rollout a new feature first to low-risk clients, to those who've signed up to a beta testing program, or to just 10% of your client base. Then, once all goes well, you can gradually increase until you're at 100%.



And if you detect a bug or get floods of negative customer feedback? Simply switch it 'off'. Even better, if your platform has a 'rollback' KPI functionality, you can set up conditions where this is done automatically.

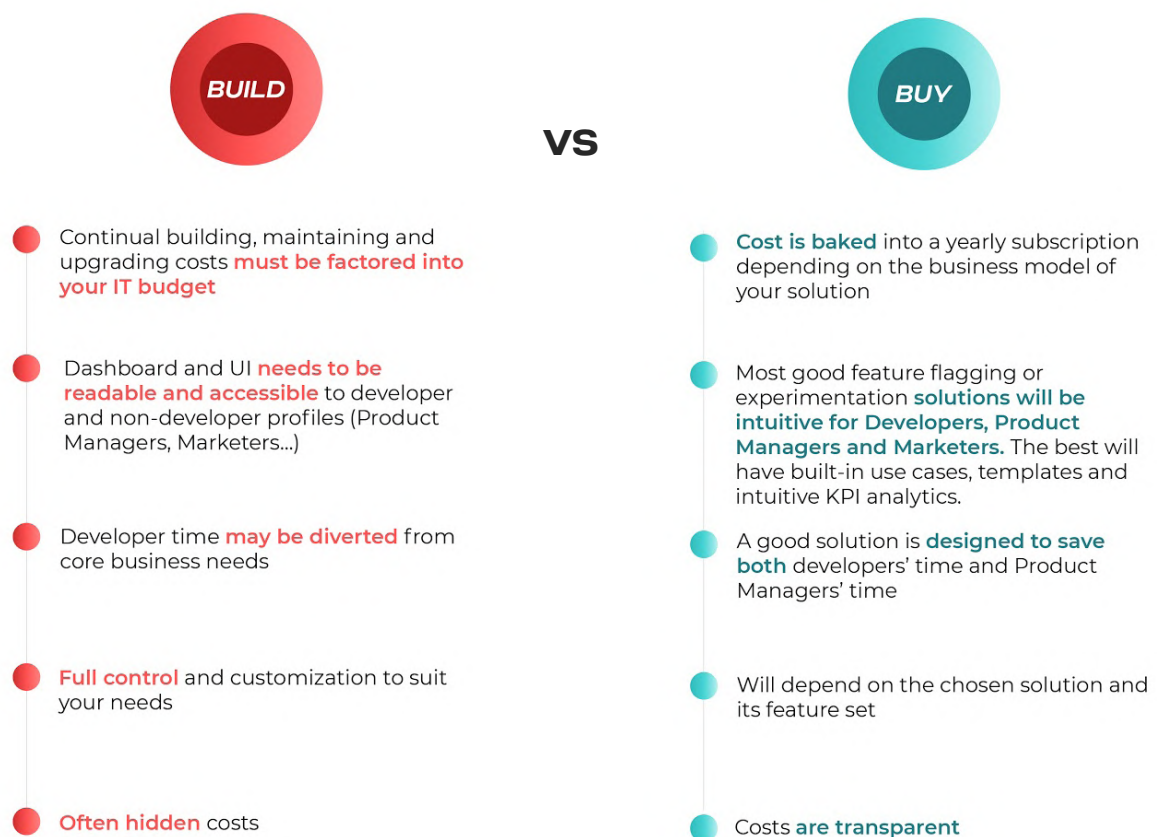
Progressive rollouts clearly have advantages when it comes to mitigating risk. They are also an opportunity to test in real conditions instead of a staging area. Indeed, it's becoming more and more difficult to create a truly realistic QA environment, and testing in production is one way to be sure to thoroughly vet your code.

AB Tasty

The customer experience optimization company

Building vs Buying, and How to Choose a Solution

At this point you might be well convinced of the benefits of using an experimentation platform or feature management solution. The question now is, should you build your own in-house, or just buy one of the many existing offers out there? There are pros and cons to each:



AB Tasty

The customer experience optimization company



About AB Tasty

AB Tasty is the fastest-growing provider of AI-powered experimentation, personalization and product optimization solutions, helping businesses drive more conversions and revenue on all their digital assets. We provide marketing, product and technical teams with an easy-to-use customer experience optimization platform, allowing them to turn website or mobile visitors into customers, subscribers or leads, while delivering outstanding user experiences. AB Tasty's customer list includes Ashley Furniture, Kiehl's, Sephora, USA Today, YSL Beauty, as well as more than 900 global enterprises.

To learn more, please visit www.abtasty.com

AB Tasty

408 Broadway - 10013 NY, New York
+1 (646) 968-0589 - usa@abtasty.com - www.abtasty.com