# Kameleoon

# PBX

# ASSESSMENT REPORT

**DEBORAH O'MALLEY**
Founder, GuesstheTest

February, 2026

# Kameleoon

# Table of Contents

# Kameleoon
# Executive summary



## How well does Kameleoon's PBX actually work?

Kameleoon's Prompt-Based Experimentation (PBX) is an AI-powered execution tool designed to help testing teams build A/B test variants directly on live websites using natural-language prompts.

Through moderated usability studies with experienced experimenters, PBX was evaluated to determine whether it could meaningfully accelerate real-world experiment execution.

The results showed that, when tests were well-scoped and execution focused, PBX consistently produced usable first-pass variants 5–10× faster than traditional manual design and development workflows.

In practice, this outcome means PBX can reduce common design and development tasks, such as copy updates, layout adjustments, or building simple UI components, from hours to minutes.

However, the findings also indicate that PBX currently performs best as an "execution accelerator." It's highly effective at generating a strong first pass that teams can refine, but at the time of this research report, PBX did not yet function as an ideation engine or autonomous build system. However, the platform has expanded to include an ideation tool, and will let you ideate and build tests in just a few licks

Strong experimentation expertise is still required to define hypotheses and create variant concepts. Design and developer review also remain important for ensuring visual fidelity, architectural consistency, and support for complex logic, highlighting several areas where PBX can continue to improve.

This report surfaces specific optimization opportunities, particularly around reducing first-time onboarding friction and improving prompt guidance. With these targeted improvements, PBX has the potential to become a dependable, high-leverage component of modern experimentation workflows.

# Kameleoon
# Usability assessment

## Report overview

[Kameleoon's Prompt-Based Experimentation (PBX)](#) is an AI-powered testing execution tool that helps experimentation teams build A/B test variants directly on live websites using natural-language prompts.

PBX is designed to accelerate the design, development, and build phases of experimentation.

This report evaluates whether PBX meaningfully accelerates real-world experiment execution for experienced teams.

Specifically, the study examines whether PBX can change the day-to-day mechanics of getting experiments live by making execution faster and easier, while reducing reliance on scarce design and development resources without sacrificing control.

## The key takeaway ------------------------------------------

For well-scoped tests, PBX can significantly reduce time to variant.
Across sessions, participants were able to produce usable first-pass variants **5–10× faster** than through manual design and development workflows.

However, PBX does not (yet) eliminate the need for human expertise.

Strong experimentation judgment remains necessary to define hypotheses and variant intent, and light design or developer review is still important for consistency and complex logic.

# Kameleoon
# Assessment of PBX performance

## PBX core strengths

PBX generates experiment code directly into usable visual variants on live webpages, significantly shrinking the gap between idea and execution.

What typically requires hours of coordination, development, and QA can often be reduced to minutes of PBX execution followed by light review.

PBX is most effective when optimization updates are clearly stated, constrained in scope, and focused on execution rather than creative judgment.

In particular, it excels at:
- Structural and content changes
- Adding or removing components
- Copy edits and Call To Action (CTA) updates
- Creating first-pass variants for refinement

In these cases, PBX reliably gets teams most of the way to a shippable variant quickly.

## Where PBX still struggles

PBX begins to show limitations when tasks require judgment rather than execution.

These areas include:
- Visual design and stylistic polish
- Ambiguous or open-ended prompts
- Complex layout logic or single-page application edge cases

At the time of this report, PBX did not have ideation capabilities. Since testing, PBX has expanded to include ideation, and will let you ideate and build tests in just a few clicks.
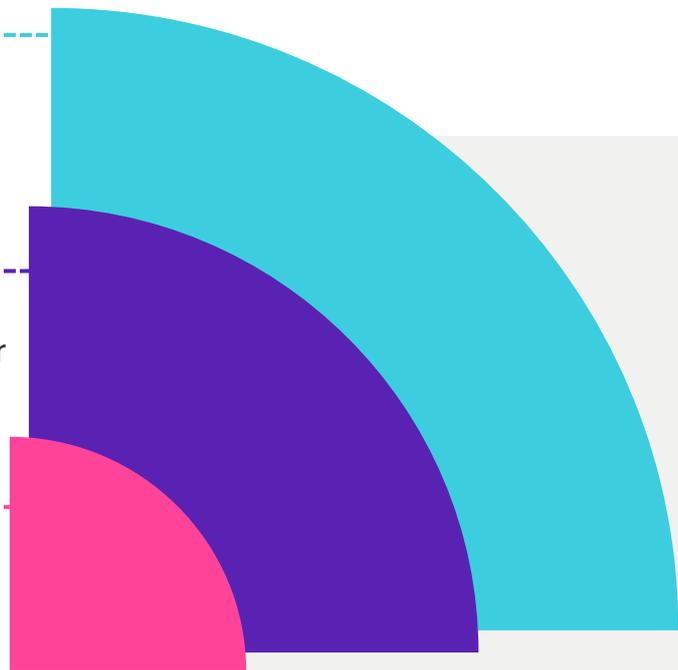
## Kameleoon

# PBX performance summary

## Strengths

- Making structural and content changes quickly with fair accuracy

- Adding or removing components and creating first-pass variants for refinement

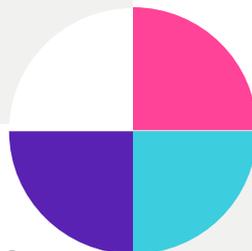- Doing quick copy edits and Call To Action (CTA) updates

## Where PBX can still improve

Stylistic polish and production-ready tweaks*

Ambiguous or open-ended prompts

Complex layout logic or single-page application edge cases

Tasks require judgment rather than execution

*Kameleoon recently announced a Figma integration for PBX that may alleviate this concern.

# Kameleoon

# What these findings mean for experimenters

For most experimentation programs, execution is the primary bottleneck.

Even small updates such as headline changes, layout adjustments, or modifications to lightweight components, often require significant coordination across design, development, and QA, causing tests to stall in backlogs. PBX helps close this gap.

When optimization intent is clear, PBX compresses hours of execution effort into minutes and produces variants that are immediately testable.

However, to support adoption at scale, PBX must consistently respect design systems and provide clearer recovery paths when outputs don't meet expectations.

Improved prompt guidance and clearer signals for when human review is needed would further strengthen trust.

**Closes the large gap in testing abilities**

**Compresses hours of execution**

**Produces immediately testable variants**

# Kameleoon

# Report scope and methodology

## Overview and participants

This report examines how Kameleoon's PBX performs in real-world experimentation workflows assessing where it already delivers meaningful value, where additional guardrails are needed, and how experimenters can begin using it as a dependable part of their experimentation stack.

Findings are synthesized from five moderated usability sessions with experimentation practitioners, including CRO strategists, UX designers, and front-end developers.

Consistent with usability research guidance from NN/G, five sessions were sufficient to uncover the majority of major usability issues.

All participants had prior experience designing, implementing, and QA'ing A/B tests using both code-based and visual experimentation tools.

## Research structure

Each one-hour session followed a consistent format:
- Moderated walkthrough and exploration of PBX
- Execution of real optimization tasks
- Comparison against participants' existing manual workflows

Participants tested PBX on websites they already knew well, including SaaS marketing sites, product landing pages, and content-heavy sections, minimizing novelty bias.

# Report scope and methodology, continued

## Prompt content

Across sessions, test ideas naturally clustered into four categories that mirrored real experimentation backlogs, including:

- **Content and copy**
  - Rewriting headlines and subhead copy
  - Simplifying explainer text
  - Updating CTA labels and button copy

- **Layout adjustments**
  - Converting grids into horizontal carousels
  - Creating side-by-side video and text layouts
  - Reordering sections to improve scanning and flow

- **User interface components**
  - Adding or removing components
  - Implementing carousels or accordions
  - Adjusting pagination indicators or navigation dots

- **Promotional elements**
  - Creating exit pop-ups
  - Displaying discount offers and coupons

## Instructions and top tasks

After selecting a site and optimization update to prompt, participants were given open-ended instructions and asked to:

- Estimate how long similar changes would typically take to design, develop, and QA
- Assess whether PBX violated or enhanced design or styling
- Suggest recommendations for optimizing PBX in future iterations

# Findings

## Typical design, development and QA time

Participants were asked whether they had built similar variants before and how long those builds typically took under real production conditions.

**Historical experiment execution time**
Across roles and organizations, average estimates were consistent:

- Simple copy or structural edits: *30–90 minutes*
- Layout changes: *60–120 minutes*
- Carousels or UI components: *1–3 hours*
- Pop-ups with logic and QA: *2–4 hours*
-

These estimates included coding, QA, and browser testing, reflecting realistic production conditions to fully design, development, and implement an experiment.
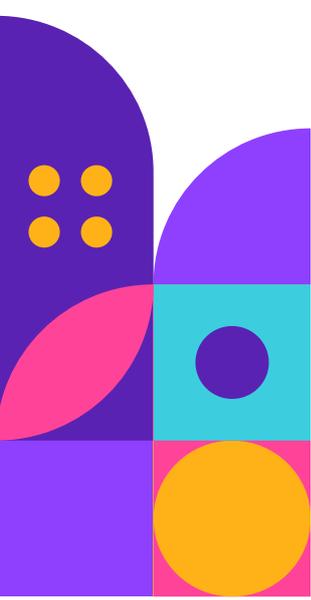
**PBX execution time**
Using PBX, participants were typically able to generate first-pass variants within minutes:

- Simple copy or structural edits: *2–5 minutes*
- Layout changes: *5–10 minutes*
- Carousels or UI components: *5–15 minutes*
- Pop-ups with logic and QA: *10–20 minutes*

While most outputs required review or light refinement, the reduction in time to a testable variant was consistent across sessions.

As a result, **PBX helped experimenters reduce execution time by 10–15x**, significantly increasing experimentation efficiency.

# Findings, continued

## Variant execution time comparison

| Variant type | Manual build | PBX | Speed gain |
|---|---|---|---|
| Copy edits | 45 minutes | 3 minutes | 15x |
| Layout tweaks | 75 minutes | 7 minutes | 10x |
| Carousel | 120 minutes | 10 minutes | 10x |
| Pop-up | 180 minutes | 15 minutes | 12x |

The chart below visualizes manual build time versus PBX build time for common experiment variants.

As shown, PBX consistently compresses execution time across variant types, shifting design, development, and QA work that typically takes hours into a matter of minutes:



Manual Build vs PBX Build Time

# Kameleoon
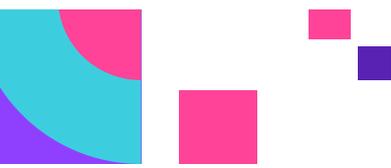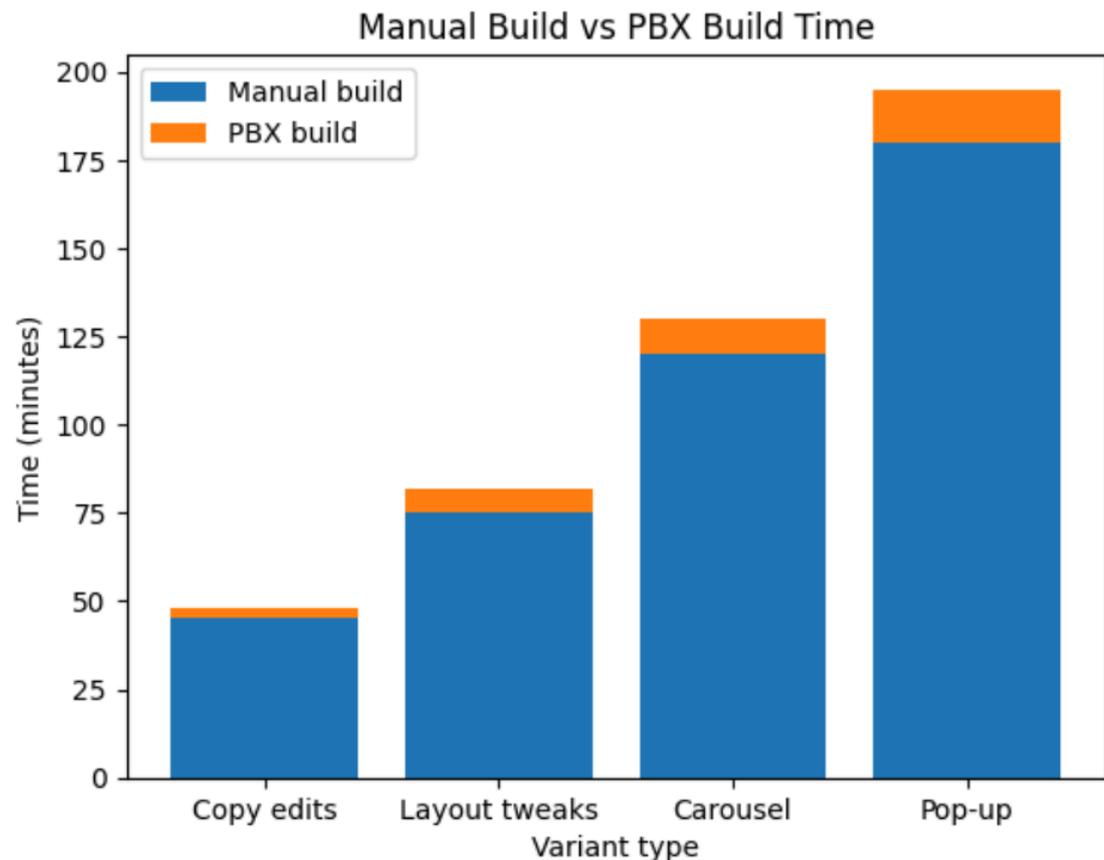
# Assessment of output quality

## Strengths

Participants consistently observed that PBX could:

- Identify large sections of the page correctly
- Produce functional, readable code
- Reduce the cost of trying new ideas
- Diminish the time required from idea to execution
- Help all experimenters turn test ideas into reality

A recurring comment across sessions was that **PBX "gets you about 90% of the way there."** The remaining effort was typically refinement, not rework.

## Correct page section identification

Across sessions, one pattern quickly became clear: PBX was generally able to understand where it was making updates on the page.

In most cases, it identified the correct sections to modify rather than targeting the wrong element or applying changes that were too narrow.
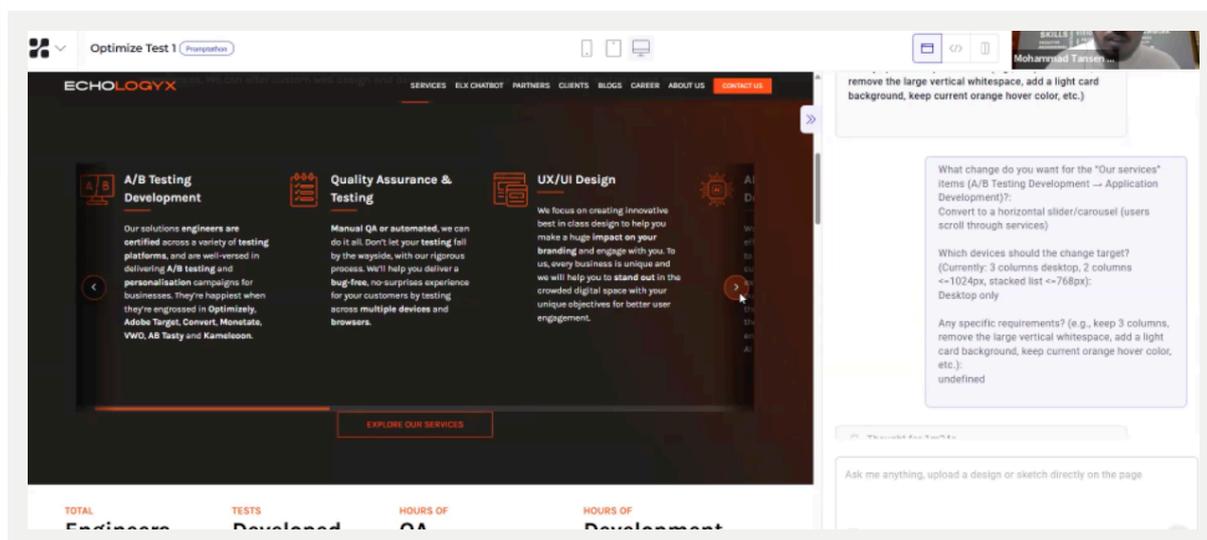
This capability was especially noticeable on content-heavy pages, where PBX was often able to distinguish between primary sections, supporting content, and repeated components.

For example, when one participant tried to turn a grid into a horizontal carousel, PBX successfully targeted the correct section, preserved the surrounding layout, and produced a working carousel structure without disrupting the rest of the page.

While the output still required visual refinement, the core structure was in place. Here's an example of the output:

## Functional, readable code

From a code perspective, PBX outputs were consistently functional and readable. Participants could clearly see what had changed without needing to reverse-engineer the underlying logic.

While the generated code wasn't always production-polished, it was sufficiently clear for a developer to step in, make a small number of targeted adjustments, and proceed with testing.

This finding aligns with a recurring observation across sessions that PBX, "gets you about 90% of the way there." The remaining effort was typically limited to visual polish and edge cases rather than substantive rework.

For teams with limited development bandwidth, reaching that 90% quickly can be the difference between a test shipping and one staying stuck in the backlog

## Enables trying new idea quickly and cheaply

Several participants observed that PBX made it easier to explore changes they might normally hesitate to pursue.

Building a variant no longer felt like a multi-hour commitment involving tickets, coordination, and development dependencies.

# Kameleoon

## Enables trying new ideas quickly and cheaply, continued

As a result, participants reported being more willing to test smaller ideas, edge cases, or secondary hypotheses.

Even when the output was not perfect, PBX helped reduce both the psychological and operational barriers to experimentation.

## Shortens the feedback loop between idea and reality

Rather than debating variants in the abstract, several experimenters noted that PBX allowed them to see a concrete version of an idea take shape almost immediately.

This rapid transition from concept to testable variant was viewed as especially valuable for decision-making.

As one participant noted, PBX helped "shorten the experimentation cycle by reducing back-and-forth and by allowing the team to decide more quickly whether a test idea was worth refining, shipping, or discarding."

## Lets non-developers execute test ideas

For experimenters without hands-on front-end skills, several participants noted that PBX created a way to translate test ideas into tangible variants without relying on design or development support.

While developer review was still viewed as important for ensuring correct rendering and behavior, PBX gave non-technical practitioners with a stronger starting point and greater agency in the execution process, particularly for routine or lower-risk tests.

For more technical users, PBX served as a practical starting point for editing and refining code, rather than replacing development workflows altogether.

# Kameleoon

# Assessment of output quality, continued

## Weaknesses and friction points

While participants saw clear potential in PBX, several recurring usability and workflow issues introduced friction, particularly for first-time users.

These challenges didn't undermine the core value of the tool, but they did affect confidence, ease of use, and perceived reliability during testing.

The primary friction points identified included:

- Onboarding and setup confusion
- Lack of clear prompt guidance
- Confusing system feedback and indicators
- Unclear failure recovery paths

## Onboarding and setup confusion

Every participant encountered some degree of friction during account setup and initial access.

Login and password requirements weren't always clear, and error messages didn't consistently indicate what needed to be corrected.

Several participants also experienced confusion during the browser extension installation process. There was no clear confirmation that the extension had been installed successfully.

# Kameleoon

## Lack of clear prompt guidance

Although all participants entered the session with a general idea of what they wanted to change, only one participant had pre-existing, well-formed AI prompt instructions.

Participants did not initially understand how to effectively translate their optimization intent into instructions for PBX.

This friction stemmed largely from the prompt box itself. While the open-ended prompt box provided flexibility, it offered little guidance on what constituted an effective prompt for experimentation.

## Feedback, status, and system clarity

While participants were generally not concerned by the absolute amount of time required to generate a variant, unclear system feedback created uncertainty during the process.

Participants also noted that, although they understood manual development work often takes time, the estimated time remaining displayed by PBX felt long relative to expectations.

Several suggested that showing real-time indicators or incremental visual feedback during generation would make waiting feel more justified or transparent; this feature now exists at the time of publishing.

## Unclear failure recovery paths

Generally, final outputs didn't fully match the participant's initial mental model. In two out of five sessions, the prompted variant didn't appear at all. This led to uncertainty that was compounded by a lack of guidance on whether to refine the prompt, restart the process, or seek assistance.

This lack of clear recovery paths introduced friction into the process. Even when PBX performed well earlier in a session, these issues left participants uncertain about how to effectively use the tool without support.

Investing in explicit recovery guidance, clearer error explanations, and safe iteration paths would likely have a positive effect on user confidence and long-term adoption.

# Kameleoon
# Summary

## Strategic fit in the experimentation workflow

PBX is still early, but it already demonstrates capabilities that many competing tools have not yet operationalized.

In its current form, PBX works best when several conditions are met:

- Design intent is clear and expressed through a specific, concrete prompt
- The optimization update allows some flexibility in execution
- PBX is used to generate a first-pass variant, with refinement handled afterward, as needed

When used in these ways, PBX fits naturally into real experimentation workflows. It accelerates execution without attempting to replace the judgment and experience of experimenters, designers, or developers.

## Final takeaway

PBX functions as a force multiplier for experimentation teams.

When applied thoughtfully, it increases execution velocity, lowers the cost of building variants, and enables more tests to ship.

Its value compounds when paired with strong CRO strategy, allowing teams to spend less time getting experiments built and more time learning from results.

Curious to try PBX for yourself? Give it a whirl. Try PBX today for free: **https://login.kameleoon.com/app/sign-up-pbx**

# Kameleoon  GUESS? theTEST ☑

# Thank You!

Please feel free to reach out to Deborah O'Malley, Founder of GuessTheTest with any questions or comments

## Contact Information :

📞 613-853-3526

✉️ deborah@guessthetest.com

🌐 www.GuessTheTest.com

💬 @GuessTheTest