

Formsort








by Head of Product

Form Builder

Details

Review Date	12/21/2023
Purchase Date	Q1'23
Implementation Time	1 week
Product Still in Use	Yes
Purchase Amount	N/A
Intent to Renew	100%
Review Source	Vendor

Product Rating

Product Overall		4.5
Use Case Fit		5.0
Ease of Use		4.0
API		4.0
Integrations		4.0
Support		5.0
Value		5.0

About the Reviewer

N/A

Reviewer Organization

- Virtual-First Provider
- Physical Therapy
- Women's Health

Reviewer Tech Stack

- Google Sheets
- Amplitude

Other Products Considered

- Typeform

Summary

- Product Usage:** The product is used for experimentation at the top of the sales funnel to increase service bookings and provides a platform for running tests without extensive engineering work.
- Strengths:** The product has strong capabilities in handling complex conditional logic, allowing for use of code in creating calculated variables that drive conditional logic, offering a higher level of customization, and providing excellent customer support.
- Weaknesses:** There's a learning curve in getting acquainted with the product and its functionalities and difficulty in saving drafts.
- Overall Judgment:** Overall, the product is highly beneficial for growing companies with limited engineering resources, despite minor UI drawbacks and a slight learning curve.

Review

Today, we're talking about Formsort and how it's being used at your company. Before we jump into that, could you give a brief overview of your company and your role there?

I lead product and technology at a physical therapy (PT) company offering both in-person and virtual PT. We have several in-person clinics, and we launched nationwide telehealth in 2023.

What core business problem were you looking to solve with a tool like Formsort?

We started with two core business needs that we were looking to solve with Formsort. Initially, our main goal was to drive experimentation at the top of our sales funnel in order to increase the number of people booking our services—we wanted to optimize our funnel.

We'd been using Typeform, but it had limitations when it came to experimentation, branding, and customization, so it wasn't able to scale with our complex business requirements. We needed a flow-building solution that was HIPAA-compliant, allowed us to apply our business logic in an intelligent way, and provided a platform that allowed us to run experiments and variant tests without requiring much engineering work.

Through referrals and conversations with other product leaders in the digital health space, we discovered Formsort and decided to give it a trial. It has been a fantastic product selection for us.

You mentioned several core requirements you were looking for as you moved away from Typeform. Did you have any additional requirements when evaluating your new form builder?

We definitely wanted something that was highly extensible and didn't require much engineering effort. Brand customization was also a major requirement for us. Additionally, integrations were crucial; we wanted the ability to easily track flows, gather analytics, and seamlessly integrate the data collected through the flows into our CRM to start fostering relationships from the top of the funnel onwards.

Formsort made this integration process quite easy. While we haven't fully explored all the available integrations, there are several out-of-the-box options and others that are straightforward to build.

Did you consider any other tools or vendors as part of your evaluation process?

We were already using Typeform, so that was our baseline, and we knew that that wouldn't meet our needs. The alternative would have been for our engineering team to build the flows themselves, but when we considered all the testing and experimentation we wanted to do, it didn't make sense for our team to spend time and resources on building flows from scratch. We wanted to empower our product and marketing teams, who are mostly non-technical folks, to be able to run these experiments on their own without relying on engineering. This led us to prioritize finding a third-party tool that could provide the necessary capabilities.

How do Typeform and Formsort's pricing models compare?

Formsort is frankly a lot more expensive than Typeform, but that's partly because Formsort is HIPAA-compliant, which was a core requirement for us. Even if we had opted for the non-HIPAA compliant version, Formsort is still more expensive than Typeform, but it's a far superior product overall.

It sounds like you went through a pilot as part of your evaluation process. Can you tell me more about your experience with the sales and onboarding process?

We started off with a trial of Formsort, where they set us up with an account so we could start exploring and experimenting with the product prior to making a purchasing decision. We were also assigned a customer success manager, with whom we maintain a joint Slack channel. That allows us to easily reach out whenever we need assistance.

The support team, in general, is very responsive. They have an Intercom integration, and they're good at responding to any questions and concerns that arise. There is a little bit of a learning curve with Formsort, and our customer success manager was instrumental in getting us onboarded. This was especially helpful for our marketing team members, who didn't have quite as much experience working with technical products. However, they quickly got up to speed and have been operating smoothly with Formsort for quite some time now.

Can you walk me through the specific use cases and stakeholders within your team? Who is using it, how are they using it, and how does it fit into their workflow?

All of the flows we've created to date are external-facing. We use Formsort for three main use cases. The first one is our booking flow. When users visit our website and click "Book Now," they are guided through a booking navigator that we've built using Formsort. This helps prospective patients find the right location and service based on their needs. All of that logic is intuitively handled through Formsort, which provides a smooth onboarding experience.

Another use case is the quizzes that our marketing team has developed. These are more top-of-the-funnel and educational in nature; they help prospective patients understand what our company offers and how our services can potentially help them, so they help educate users and generate interest.

The third use case is the Info Hub. We operate nationwide with different insurance networks and rates, which vary by market and location. Presenting all this information on a single webpage would be overwhelming. To address this, our marketing and operations team has created a flow in Formsort called the Info Hub. Depending on the location users select and the region they're in, the relevant information is displayed to them. It's less about conversion in this case and more about how we can most intuitively serve up information to our users.

Marketing owns all of the quizzes—we currently have three quizzes in place. The booking flow is owned by the Product team, and the marketing team also handles the Info Hub flow.

When you mention that they own it, are they able to fully optimize and make adjustments to the logic as necessary?

Yes, they are responsible for designing, implementing, launching, and monitoring their flows.

You mentioned that one of the key advantages of Formsort is its ability to handle complexity and customization. Can you tell me more about Formsort's capabilities and standout features?

The biggest difference I've noticed between Typeform and Formsort is the level of complexity that can be achieved with conditional logic. Typeform is very basic when it comes to building conditional logic; it's all handled through their UI, where it's mostly "if this is selected, then that." This format becomes very cumbersome when dealing with hundreds of combinations that can lead to a particular path.

On the other hand, Formsort provides a UI for building conditional logic, which is great for non-technical users, but it also allows for the use of code to create calculated variables that drive the conditional logic of each step. With Formsort, we can call on specific information collected during the flow and write functions that process that information as users progress through the flow. This approach offers a safer and more comprehensive way to handle conditional logic at scale.

Writing the code is not overly difficult, and tools like ChatGPT have made it even easier. Overall, it's made managing and scaling the logic considerably easier.

How would you assess Formsort's strengths and weaknesses?

Formsort is an incredibly powerful product that allows you to achieve a great deal and create a native and intuitive experience without much, if any, engineering work. This is very powerful for growing companies with limited engineering resources.

As for weaknesses, there is a slight learning curve; it did take some time to become familiar with the product and its functionalities. Additionally, there are a few minor UI elements that can make navigation a bit challenging at times, but the Formsort team is already working on addressing these issues.

Another downside is that saving drafts can be difficult; you have to decide whether or not to commit to specific changes. However, Formsort does offer different environments, which allows you to push flows to a staging link so you can thoroughly test them. In contrast, Typeform only has production or development environments, making it harder to roll back changes if needed. The presence of different environments in Formsort makes it a lot safer to experiment.

Has Formsort been stable and reliable in your experience?

In terms of stability, we have not experienced any issues with Formsort. There have been very few instances, definitely less than five, where I encountered bugs, most of which were related to conditional logic and occurred in very uncommon use cases. Whenever I came across a bug, I would record a video and share it with the support team. They were very responsive and usually resolved the issue within a few days.

You mentioned that extensibility was one of your requirements. How have you built on top of Formsort using the API, and which out-of-the-box integrations have you implemented?

One of our main out-of-the-box integrations is with Google Sheets, which serves as a backup of all the response data. This allows us to quickly perform pivot tables and perform basic analyses when needed. However, most of our analysis is done using Amplitude, which we've also connected to Formsort using an out-of-the-box integration.

The great thing about Formsort is that we can deselect the option to send personally identifiable information when appropriate. This feature allows us to mask sensitive data when running analytics on a platform; it gives us the ability to deidentify the data or simply hide it and not send it to Google Sheets or Amplitude.

The third major integration we use is the webhooks feature. As we collect information from the flows, we immediately send certain data, such as email addresses, to our CRM using webhooks. This integration has been easy to build, and each time we want to send new information from Formsort to the CRM, it takes less than half an hour of work from our engineering team. There's a little bit of engineering work required, but it's a very straightforward process.

How would you describe the quality of the developer experience and the documentation provided by Formsort?

We haven't encountered any issues with the engineering work. The documentation provided by Formsort is quite good; they have Loom videos embedded throughout their documentation, and they also include various samples of integrations and flows, which makes their documentation highly relevant and easy to learn from. Whenever I get stuck, the support team is very responsive and helpful.

From a customer standpoint, is the booking form something that customers use only during their first interaction, or is it a flow they utilize every time thereafter as well?

The booking flow is primarily designed for first-time users. If someone accidentally revisits the booking flow, we direct them to a follow-up process that is different from the initial booking flow. We don't have a specific flow associated with handling these cases, so across the board, the primary use cases for Formsort are focused on top-of-funnel interactions.

I'd like to circle back to something you mentioned earlier, which is experimentation. How has Formsort enabled and supported your experimentation efforts?

Formsort allows us to create multiple variants within our flows and assign different weights to each variant. This is incredibly valuable for experimentation. Because we can keep everything within the same flow, we can use Amplitude to compare the performance of each variant by segmenting the data by variant.

The process of creating multiple variants is straightforward: there's a duplicate variant button that allows us to duplicate a version within the same flow. From there, we can make the necessary changes, and the duplicated variants function independently, ensuring there are no interdependencies to worry about.

One limitation we've noticed is that if there are changes to the business logic, such as opening a new location, we must replicate those changes in each variant individually—the independence of the variants is both a pro and a con. Formsort is addressing this limitation with a new feature, currently in beta, called the content library, which allows us to create shared modules that can be applied to multiple flows. This means that when we make a change in one place, it will be reflected across all relevant locations. One of my quality-of-life projects this week is to move some of those shared components into the content library. This will significantly streamline the process, especially when dealing with changes that impact multiple areas.

What has your experience with Formsort's account management and support teams been like?

They've all been fantastic and very responsive. I even had the opportunity to meet with Ben, their CEO, recently and got a preview of their product development pipeline. It's clear that they're very eager to continue building their product and are open to receiving feedback from customers. So, overall, I have no qualms.

Looking back, do you think you made the correct decision in choosing Formsort?

Yes, I believe our decision to move forward with Formsort was the right one.

You mentioned a number of weaknesses. If you were to highlight any of these for the product team as priorities for them to resolve, what would you say?

I think the most pressing priority is the need for common modules that can apply to different variants or flows that can be controlled from a central point. I was very pleased that they turned on that beta feature for me after our conversation.

The second priority would be the calculated variables, which is a feature that allows us to write custom code to affect the function of the conditional logic. This is a very specific point, but we're unable to save and share drafts of these variables. It would be helpful to have the ability to save drafts and manage them across variants; I think this affords Formsort the opportunity to create a slightly more robust builder.

Another area for improvement is the design responsiveness across different viewports, such as mobile and desktop. It would be great to see how the design changes across different screen sizes. While discussing this with Formsort, I was informed about the possibility of injecting CSS to achieve different designs for different screen sizes, although I haven't had the opportunity to experiment with it yet.

Do you have any advice for buyers who might be looking for a similar tool?

Choose Formsort. I don't mean to sound like a salesperson, but I feel that it's a highly effective and powerful product that can substantially improve your conversion rates or other core metrics without requiring much engineering work. What's great is that anyone can be trained to use it, so it really democratizes this function.