

Athenahealth

by Senior Product Manager

Telehealth Platform

Claims Management








Electronic Health Record

Headless EHR

Details

Review Date	07/12/2023
Purchase Date	Q2'21
Implementation Time	N/A
Product Still in Use	Yes
Purchase Amount	N/A
Intent to Renew	N/A
Review Source	Elion

Product Rating

Product Overall		3.0
Use Case Fit		3.0
Ease of Use		2.0
API		2.0
Integrations		2.0
Support		3.0
Value		N/A

About the Reviewer

Implementation Team Product Oversight

Reviewer Organization

Specialty Practice Women's Health

Reviewer Tech Stack

Phreesia ActiveCampaign Zocdoc

Other Products Considered

Epic Maternity Neighborhood

Oracle Cerner

Summary

- **Product Usage:** The user leveraged Athena for various operations excluding patient scheduling (handled by Phreesia) and patient care management tasks (handled by ActiveCampaign).
- **Strengths:** Athena offered an excellent sandbox environment for testing and its user interface was considered more intuitive compared to competitors like Epic.
- **Weaknesses:** The user found it challenging to work with Athena's APIs due to incomplete documentation, and getting support with these issues was difficult.
- **Overall Judgment:** While Athena was suitable for the company's needs as a brick-and-mortar practice, the user questions the additional value gained from building custom apps on top of it.

Review

Could you give just a brief overview of the company and your role there?

The company is a women's health provider that functions largely like an OB/GYN practice, but is tech-enabled and provides 24/7 hospital care. I was the product manager there, so I was in charge of anything to do with technology – digital vendor selection and purchasing, or development and implementation of digital tools.

So you ended up purchasing Athena. When did you go through that process? And for how long were you using it while you were there?

It was already purchased. I joined in summer 2021 and they had just purchased it. The real reason that they decided to go with it was because it was thought to have the best APIs to use, and there was a strong desire to build a patient-facing experience that was custom to our company off of it. For the most part, it really came down to that. There are some maternity-specific EMRs, but they just didn't have the capabilities that Athena had when it came to APIs.

It sounds like you were considering some more traditional and maternity-specific EHRs. I'd be curious to understand whether Athena was able to match or compensate, from a workflows perspective (i.e., having specialty-specific workflows)?

It was tough. There was a provider whose full-time job was doing Athena workflows and getting it set up two years before the first practice actually opened. I spent a lot of time with her and with our Athena account manager to get it set up to be super specific for maternity care – for the type of visits, the fact that a lot of visits end up being canceled and rescheduled, etc. I would assume that every EMR has that problem, because every practice is pretty unique in terms of how they run things. But this provider had worked at other practices in the past, and so she really replicated what worked and what didn't work in Athena.

What were some of the actual workflows that you were going through in Athena? Did you do scheduling there or use them for charting clinical notes, all the way through billing?

We used them for almost everything, with a few exceptions. We used Phreesia for patient scheduling, which allowed patients to schedule visits. We used it for that and for intake forms. The other thing we did is that we had a patient care manager who did a lot of check-ins and calls that were supporting care, and that really wasn't well-supported in Athena. So we used ActiveCampaign, primarily their CRM functionality, to manage those pieces.

How did you carve out the areas where you used Athena versus the areas where you supplemented with other products?

For scheduling, we wanted someone to be able to come to the scheduling widget, and, based on the type of visit, go to any of the clinicians. They didn't allow for that, but the feature was critical because a lot of our providers (this might be obvious) were not available very often, and it would have been really challenging for people to find an appointment. That is when we decided to look for options outside of Athena scheduling and looked at Phreesia. I think that came with the intake forms component. And so we used that as well.

To use Phreesia for scheduling, did you also need to use their intake forms?

We might not have had to, but it did make it easy. It wasn't necessarily something we were looking for. It really had more to do with self-scheduling being such a critical piece of tech-enabled care. Nobody these days wants to call, obviously. And so, having something that was really going to meet our needs, especially as a new practice, where a lot of it was going to be people just making a quick appointment after seeing us in some ad. That's what led us to look through it, and I forget what the other option that we had looked at was called, but I think they all basically do the same thing.

Did you consider other vendors, like Acuity or Zocdoc?

We needed something that integrated well with Athena, and for some reason, I feel like Acuity did not integrate well. We used ZocDoc but more for marketing purposes than for scheduling. We integrated them with Athena, not with Phreesia.

And you also mentioned using ActiveCampaign.

Theoretically, we probably could have used Athena for this, but this was not for care that we were providing – it was a service outside of care. A lot of it was scheduled check-ins and tasks. It was a lot easier to set that up for something that was changing more frequently. I mean, I think all EMRs are just tough to configure. And so this was a lot easier to configure. I think the original thought was that we would use Athena, but we did have ActiveCampaign for our marketing emails. And then we decided, as it was a struggle to put together the workflows we were looking for in Athena, that we could use ActiveCampaign. It was a bit easier to modify.

From the provider's perspective, when they were using Athena, what features worked well? What features were lacking from their perspective?

I'll tell you what I experienced also, because it impacted me. Messaging between providers and with patients; that feature was quite confusing and a little stressful because the way you did it, it was like one long thread where you could pass it off to someone else but it wasn't really clear where the message was coming from. I got a lot of questions because we had built a patient app off of the APIs, and so, there was a lot of concern around what was showing up or not for the patient. That was a challenging piece. And the routing of that was challenging, e.g., what box is the message going to go to?

There was also something really specific with the ultrasound technology that the ultrasound techs were using that didn't play well with Athena. And so, I think we were manually uploading instead of being able to just send files. I think that a lot of other practices were able to send across files, but we were manually uploading them. I don't know if they're still doing that, but that was definitely a challenge.

Were there any features you ended up not using because they weren't functional for you?

I think the main one was self-scheduling. The plan was never to use something else. Having now worked in places where I'm picking the whole tech stack, the biggest thing is that what people say they do and what they actually do are a little bit different. And so, we had asked ahead of time how scheduling works. You also make a lot of assumptions, like, "Obviously, it should allow you to do this specific thing," and then you start to use it and you realize, "Oh, this didn't actually work like I expected it to."

Everyone also makes a lot of assumptions about what the word integration means, and thinks because they integrate, it means that every field can go back and forth between two different tools. That is just never the case. It's always

a subset of things. And maybe it's bi-directional, maybe not. For the tools that were purchased before I joined the company, there were assumptions that they would all integrate; we were told they integrate. But no one dug that layer deeper to ask what that means or what information can be passed and in what format. And so they get stuck with tools that don't really do what they want them to do. But now they're bought in for, like, three years.

You mentioned you had built using Athena's APIs and that they were a really important part of the decision process. How did that go?

I think a lot of the documentation was great, but a lot of it was missing information or unclear. And so we upgraded to a very expensive package where you get support from services. So, once a week, we met with consultants. We had approximately 10 consultants over the years. They weren't technical, but they were supposed to help us understand what we could do and what we couldn't do. Often, they didn't know what was possible or not. They had to go back and ask their engineering team. A lot of things you'd think you could do, based on the documentation, but they pointed out we couldn't because of some nuances.

There was another health tech startup that was also using them without platform services and they were having so much trouble. So even though we had that very expensive service, it was still very challenging to make sure what we could do and what we couldn't do. For example, I worked on lab results for a very long time and the current head of product there just finished it over a year later. It should be so simple. I just wanted to display labs, the way they have it in the patient app – I wanted to do that on my side and nothing ever showed up in the right place. It was always displaced, or incorrect analytes showed up, and nobody could really help us there. But we thought it should be simple because they had already built it.

Unless you're really doing something drastically different for patient apps, I would use what they have. I would say that ours looked a lot better, UI wise, and was on brand, but for the amount of money and effort that went into it, it really didn't do anything else.

That's helpful to hear.

This might not be a popular opinion because investors want to see that you have an engineering team and that they're doing the tech component. A lot of these platform companies have already built the technology in a fairly robust way, and trying to build on top of it, build your own patient experience, may not be worthwhile. People think in the beginning that the way they build is going to be so much better, but they don't think about how long it took Athena to build the same feature, with a smaller team and without the benefit of being an insider at the company.

Going back to the feature set, I'd like to understand the billing component, and the level of integration and effectiveness of that. What were your thoughts on using Athena as an EHR and an RCM system versus just as an EHR?

At the very beginning, we had an RCM specialist and she took over all the management of it. I can probably only say that I don't think she thought it was bad. But I think she was working a lot, constantly going back and forth with Athena's people. We were always talking to them, people, across the whole experience. We were always running into issues and opening support cases, with varying degrees of success.

We've talked a little bit about integrations already. Do you feel like the integrations and the different apps were fairly robust? Were they deep integrations or did they feel more shallow?

The APIs didn't allow for a lot of the things that we wanted to do. So I think every time they'd be like, "Well, you could get an integration to do this," and it never really did what we wanted it to do. I think that's also down to the word – integration doesn't mean you're going to be able to get and pull all the data you want. We really wanted a phone system that was integrated, so you'd be able to quickly call someone from inside and none of the apps really allowed for that. It seemed so simple to us. We'd never have to write our own code for any of it. I spent a lot of time with Phreesia and there was a lot that happened between Phreesia and Athena: they had to do a lot of things on their end. It was not like, for example, Typeform and Google Sheets' integration.

Any time I've talked to anyone that has worked with Epic or any of the other EHRs, it's always the same thing. I don't think Athena is probably any worse than any of the other ones. It's just that if you're trying to do something somewhat bespoke, it can be rough.

You weren't necessarily involved in the procurement decision, but do you recall if you evaluated or considered other products to try to fit the need?

We took over a practice that was on a more legacy system, so we had to pull data from it. I think they might still be working on getting all the data into Athena. I think that process will make them never want to leave Athena. And so I think it's a sticky product – it's too hard to get out. But before, they definitely looked at typical ones like Cerner and Epic. I think Maternity Neighborhood is the maternity-specific one that a lot of the midwives and OB/GYNs really like. I think it truly came down to the API piece, because we were tech-enabled. Athena may not be great, but in comparison, they had the most open APIs. I think that's how they sell it and for the most part, they're talking to non-technical people. Without that knowledge, it sounds like Athena has what one might need.

They have expansive API docs, but it sounds like things didn't necessarily do what they needed to do, or what you expected.

The platform services support we got – I wouldn't say they were super helpful, but they were really helpful. And we were able to do the testing process for API calls on those calls. At another company where we didn't have the platform services support, we had to submit the documentation for what we wanted to do, and it was a long document. They would then approve it, then you had to get on a call with them to demo what you wanted to. You'd get the approval after you did the demo and they checked the logs on their side. For us, it was usually two weeks but it could take at least a month if you wanted to get something going. And because of the demo, you had to get it done before you could get it approved. Because we had been working with them consistently and telling them what we were doing, they could tell us any red flags they saw. I think in theory this is a good practice because you're ensuring the data is not misrepresented but it can be pretty time-intensive.

You were looking at a lot of the bigger, more legacy EHRs or maternity-specific ones. I'm curious why they didn't look at tech-first organizations that were building EHRs.

Among other things, it depends on how much you're gonna be working with, for example, huge hospitals. I think part of it also is that you want to establish yourself as reputable and trustworthy and not risky. I'm making assumptions, but if you really are standalone and don't need to worry about working with hospitals early on, it's different. One Medical, for example, built their own but they were standalone – they weren't trying to partner with big organizations from the beginning.

Looking back, do you feel like the team made the correct assessment at the time to go with Athena?

For their needs, as a brick-and-mortar practice, I think it made sense. At the same time, I think there needed to be more consideration about the marginal value added by building an app on top of Athena, versus just using what comes out of the box from the EHR. If they were buying Athena because it's a traditional EHR that has APIs, but they didn't really need to build an app on top of it, it might have been better to buy the best EHR regardless of APIs, and use what they have out of the box. But given the strategy at the time, Athena was fine.

What did you like most about the product?

I haven't really worked with any other regular health EMRs before, so I'm assuming all of them have this, but Athena had a really good sandbox environment. We could do a lot of testing in it. Obviously, it wasn't real data, which was hard with labs because you couldn't fake data, but I thought that they did a pretty good job. When we were actually trying to test some of our functionality, I wasn't super worried. Except for that one feature, messaging, we were able to replicate the exact scenarios of passing it doctor to doctor because I could create all these people. And it was pretty easy to do. So I liked that. I think people at least thought it was a little bit cleaner than some of the others. I think compared to some of the others' UIs, it was a little more intuitive. That was the general feedback that we had gotten – that it was easier to maneuver than Epic.

What was the thing you disliked the most about the product?

It was less about the product and more about dealing with the API – if it wasn't completely clear from the documentation, getting an answer was so challenging, or we never got an answer because nobody actually knew what to do. In general, the experience of working off of their APIs was pretty challenging. And I don't think we did anything that was very unique. I imagine most people were trying the same thing; someone else must have had this question before.

Is there anything that we didn't cover that would be worth mentioning?

Unrelated to Athena, figuring out the other tools in the tech stack was horrific. For example, knowing which ones were going to be HIPAA-compliant was a challenge.

Do you have any advice for someone selecting this type of product right now?

I think it's helpful to try to write out scenarios and actually have use cases in mind, and then have the salespeople walk you through how those could work. I think it's really easy to make assumptions about things that will happen, based on what the salespeople are saying. But when you get down to the nitty-gritty, you realize it does or doesn't do what you need. So I think being really specific is helpful.