Elion

Buyer Review Transcript

Product Reviewed	Acuity
Integrations Reviewed	Zoom
Other Products Mentioned	Squarespace, Salesforce Scheduler
Date	July 19, 2023
Expert Role	Head of Technology
Specialty	Tech-Enabled Primary Care
Payment Model	Fee-for-service and Value-Based Care

So today, we're talking about Acuity and how it was used at your former employer. Before we jump into that, could you give a brief overview of that company and your role there?

We were one of the first companies targeting pediatric behavioral health care. There was a three-tiered offering of a digital-only product, coaching for lower acuity cases, and then all the way up to therapy and medication management for higher-acuity cases. I was the chief technology officer there. I was one of the first employees, and we grew the team over about three and a half years, and I did everything from building out our technical tools and implementing our products to launching and scaling them with the team and company.

That makes sense. When did you purchase Acuity? And then how long were you actually using it in production?

We used Acuity the whole time I was there, so from 2020 through 2023. We talked about replacing it a few times, and I can walk you through like those decision points, but we used it from the very early days of assembling a system together all the way through a pretty large production scale.

It sounds like there were multiple phases in the lifecycle of the company where you might have used the product slightly differently. How did you use Acuity when you first acquired it, And then how did that change over time?

Yeah, so at a high level, when we were first building our tech infrastructure, we had to decide whether to go with an off-the-shelf EMR solution, which offered an integrated system, or build our own. We knew we wanted to develop our own system quickly, especially since this was during the early days of COVID. After careful consideration, we chose to create our own system and aimed to launch it as soon as possible by using as many off the shelf components as possible, but NOT using a fully integrated solution so we would have more flexibility down the road.

I collaborated with a couple of engineers I had hired and we brainstormed the tools we could assemble rapidly to achieve our goal. In just 10 weeks, we went from having no code to launching our first system that actual patients could use. It was a challenging task, and I had to figure out how to piece together tools without needing extensive integration, yet still integrate them together effectively enough to build a system without starting from scratch, particularly in terms of scheduling.

During this process. I thoroughly researched the landscape, including options like Salesforce and several scheduling tools (around eight or nine, if I recall correctly). Honestly, most of them didn't meet our requirements and fell short, especially in terms of obtaining a Business Associate Agreement (BAA). That's when Acuity emerged as the only real choice for us at the time.

What was particularly advantageous during the initial phase was that Acuity allowed members to schedule directly using Acuities UI components . We leveraged their pre-built member and clinician-facing features, integrating them with our backend using custom appointment data and webhooks. Their API was solid and well-documented, making the integration process straightforward. Moreover, Acuity's affiliation with Squarespace allowed us to sign a BAA with them pretty easily, all at a remarkably low price point. Additionally, we wanted staging and development environments, along with the ability to duplicate configurations, and for each of those environments, we paid next to nothing. Considering the BAA and the features, it was an unbelievably affordable solution for us.

However, it's important to note that Acuity's support operates asynchronously. They don't have account reps to guide you or provide real-time assistance at the lower price points. Instead, you submit inquiries and receive responses that can take minutes or even a day or more. That was the support model we had to work with. Over the ten-week period, we developed a cycle of asking questions, receiving answers, and trying out new approaches. It did take some time and involved iterative back-and-forth communication. Another aspect to consider is that the system itself is quite rigid—it performs its intended functions well, but making changes is not their priority unless you're at a high-priced, enterprise-grade level (and even then, it’s not their priority to change the platform.) Nevertheless, they were cooperative and willing to collaborate asynchronously, finding creative solutions based on the existing functionality they provided. In hindsight, Acuity was an excellent starting point for us and served us well during the initial year.

That's a really helpful introduction. So just in terms of the user workflows, were you trying to accomplish patient self-scheduling?

That wasn't our primary use case, but it turned out to be convenient because we didn't have a large staff. Initially, our main focus was onboarding and getting people signed up for their first intake appointment. We wanted the entire process to flow smoothly since our system was designed for families. It involved gathering information about parents and kids, filling out forms, obtaining consent through digial signing, and dealing with all those necessary steps. Additionally, we needed to ensure they could easily schedule their appointments.

To address this, we created a custom patient experience and integrated Acuity's widgets within that flow. It wasn't a simple "Here's a link, go schedule yourself" approach. Instead, we designed a comprehensive flow where Acuity seamlessly fit in the middle, allowing us to incorporate its features.

Got it. So, let me make sure I fully understand. Were you primarily relying on administrators to handle the scheduling once the patients had signed up and completed all the intake forms? Was that the main workflow, or did you also have a process where patients could schedule themselves after the backend processing was done?

We experimented with a few variations. The great thing was that we had flexibility in how we approached it. For instance, our front-of-house team could directly access Acuity and handle the scheduling if we wanted them to. We also had chat communications with our patients' parents, so there were times when our member support staff would simply send them the Acuity link through chat and say, "Hey, just go ahead and book an appointment here." We had a range of pre-built features and functionalities that we could put together and use in different ways, depending on the situation.

So, in general, you were able to make use of the pre-built components and features provided by Acuity, rather than having to create your own calendar elements from scratch using the API. And it seems like it worked well right off the bat. What are your thoughts on the overall composability of Acuity? In other words, how much could you embed or integrate it into your own user interface or workflows?

That aspect actually worked really smoothly. Whether we were embedding the Acuity widget directly or using the self-scheduling link that we sent over chat, or even making API calls from our backend, everything worked together seamlessly. We had the freedom to choose any of those options, and they all played nicely with each other. However, looking back, one thing to consider is that much of our experience with Acuity involved working around its limitations and finding creative solutions. At the time, we were so focused on dealing with the constraints and being inventive with our approach that we didn't spend much time thinking about how we would have ideally liked it to work or what an ideal solution would have looked like.

Got it. So, can you give me some examples of the limitations and constraints you encountered while working with Acuity?

Sure, let me share a few examples where we started running into issues. Initially, we had a surplus of available time slots from a few providers, but very few patients. However, as we began to fill up those slots, we wanted more control over our clinicians' schedules. We wanted to have specific appointment times available for certain clinicians and limit the types of appointments they could accept during those times. Unfortunately, the configuration options in Acuity were quite limited for these requirements. We also wanted to ensure that there was a designated amount of break time per day, spaced out evenly, but that was challenging to achieve within Acuity's capabilities.

One area that got particularly tricky was managing provider credentialing and licensure. We quickly went live in all 50 states, which resulted in a complex web of providers licensed in different states. Our goal was to get patients their first available appointment, but with the varying appointment rules and licensure requirements, it was practically impossible to accomplish within Acuity's constraints. As a result, we had to develop additional logic and implement subpar solutions to navigate these challenges. In fact, some of these mediocre solutions are still in place today, and it's one of the reasons we're considering replacing Acuity at some point.

Yeah, that totally makes sense. With telehealth, one of the really challenging aspects of working across state lines is dealing with licensure and ensuring that you have the right providers scheduled with the right patients in the right locations. Has Acuity been able to address that issue in a comprehensive way, or are you stuck managing your own licensure and creating your own routing algorithms?

It’s the latter situation – we’re really doing our own routing. Acuity has its own focus, and telehealth isn't their primary area of expertise. They were willing to work with us and come up with creative solutions within their existing infrastructure. However, once we reached that point, the solutions were far from optimal. Unless Acuity decides to prioritize and enhance these specific features, they will always fall short of being the ideal solution.

That makes sense. It seems like another aspect of your care model involves multiple parties with both parents and adolescents involved, right? Were there any scheduling components that became more challenging due to this particular care model?

It wasn't just more challenging; it was literally impossible. In Acuity, an appointment is strictly between one provider and one customer. Sure, you can add multiple email recipients for the appointment, but there's no concept of a many-to-one relationship at all. So, it's impossible to handle situations where both a patient and their child need to accept or be part of an appointment. We wanted to establish group appointments, but that's not supported by Acuity. These are hard limitations that exist in the current system.

You mentioned that you explored the possibility of moving away from Acuity. I'm curious about the thought process behind that decision and whether you found any alternatives that appeared to be more competitive or better suited to your needs than Acuity.

There are a few factors to consider. As I mentioned earlier, in the beginning, we paid a couple of hundred dollars a year for Acuity, which was incredibly inexpensive and an easy choice for our initial system. But things started to get tricky in a few areas. Firstly, we needed to modify our BAA due to our specific needs. It was a minor change, just a few words, to ensure proper coverage under the BAA. However, Acuity/Squarespace was only willing to make redline edits to the BAA if we upgraded to their enterprise plan. So, our costs skyrocketed from around $300 a year to a base price of $20,000 per year. It was steep, just for such a small modification. They took a hard line on negotiations, and although we ended up doing it, it left a bitter taste in my mouth. On the other hand, the higher-priced plan did offer better support, an account representative, and a higher baseline level of features.

Initially, the $100 plan sufficed since we had a limited number of users. But with the higher price, the costs became exorbitant. I need to check my notes, but I believe it was somewhere around $0.50 to $1 per appointment, even at scale. Over time, due to the limitations, we started building more internal solutions. We decided to handle our own emails instead of relying on Acuity's reminder and confirmation emails. We built our own user interface rather than using Acuity's UI. Gradually, it became primarily a backend system for our provider managers to manage provider availability and handle appointment conflicts. And yet, we were still paying that high rate per appointment, which is quite steep considering the value Acuity was providing to us over time.

We also realized that we needed features like group appointments, which Acuity didn't support. We wanted to offer more comprehensive care and explore different workarounds. For example, at one point, we couldn't retrieve the availability of 10 providers simultaneously. We had to make multiple API calls through the UI or find alternative approaches. There wasn't a straightforward way to say, "Give me the first available appointment among these 10 people."

So we were stuck with all these workarounds and suboptimal solutions, and we really wanted to implement group appointments, but it was costly and didn't make sense anymore. The thing is, even after exploring alternatives about a year ago, we couldn't find anything better than Acuity. The problem it was solving wasn't actually that difficult, so we started considering building our own system in-house if we were to move away from Acuity.

Looking back, I can say that for the first year or two, Acuity was really great. It fulfilled all our needs, and we didn't have to build anything ourselves. We had various options for utilizing it, and I would choose it again without hesitation. However, as our use cases became more complex, it became expensive and a bit of a hassle, with no way around it. Building the system from scratch ourselves would be a significant undertaking. So, we just haven't found the time to tackle it yet.

Got it. So during the procurement process, did you explore any alternative solutions? Can you recall any other options you considered and how they compared to Acuity?

Yeah, I don't remember all of them off the top of my head, but there were a few that I looked into. Some of them seemed a bit shady, like this one with only four people. I did some testing with them, and they claimed to be HIPAA-compliant and all that good stuff, but it just didn't work out. Appointments were all messed up, and even during my beta testing, I kept running into significant issues. Salesforce was another option, but it was way too complex and a whole different ball game. Then there were three other middle-tier alternatives that seemed alright, but they were still way below Acuity in terms of capabilities. When it comes to procurement, with Acuity it was a breeze. I simply logged onto their website, set up my own account in just an afternoon, and boom, I was up and running. I signed the BAA and was good to go. I didn't have to deal with an account rep, but at the same time, I didn't need their help either. It was a self-service experience, and I had our production Acuity account set up within minutes.

Yeah, that self-service aspect definitely makes a difference. I'm curious about the integration between Acuity and your EHR. Can you explain how they worked together and the relationship between the two systems?

Actually, I intentionally avoided any direct integration between Acuity and our EHR. When setting up our infrastructure, our backend system handled all the communication between different third-party systems. So we integrated with Acuity and then integrated it with our EMR, but there was no direct relationship between Acuity and the EMR.

Got it, so there wasn't a direct connection where Acuity would talk directly to your EHR. Instead, Acuity communicated with your APIs, and then your APIs would route the information to the EHR.

Exactly. The only direct interaction we had with Acuity was through the Zoom scheduling feature. It was actually quite nice most of the time. When you created an appointment in Acuity, it would automatically generate a Zoom link, and you could push that link to the provider's calendar or send it via email. So that part worked well. However, occasionally, it would break. The Zoom integration was like a plugin within Acuity, and sometimes it would disconnect, resulting in appointments going out without the Zoom link. It was a bit of a crisis to fix, but fortunately, it only took me about 10 seconds to log into Acuity and re-authenticate the Zoom plugin. Apart from those occasional hiccups, the Zoom integration worked great. It's just every once in a while, chaos would strike.

How would you describe the Acuity API's overall robustness and extensibility? Did it provide the necessary endpoints you needed, and was the documentation helpful?

Yeah, their API documentation was top-notch. As I mentioned earlier, Acuity's self-service model extended to their API as well. The documentation was great, and building integrations using their API was a smooth process. Sometimes, we did encounter limitations or peculiarities that required us to work asynchronously with their support team to find solutions. We would discover that something worked differently in the API compared to the web interface, and we had to figure out why. Once we worked through those quirks and built our workarounds, we were able to keep moving forward. Overall, the API was quite stable. I don't recall it ever going down or causing our integration to fail unexpectedly. It was a reliable and solid API.

What did you like most about working with Acuity?

The self-service aspect.

And what did you dislike most about the product?

The self-service aspect.

Is it true that our greatest strengths can also be our greatest weaknesses?

In this case, yes.

Looking back, do you feel that choosing Acuity at the time was the right decision?

Yeah, for sure. The struggles we faced later on weren't surprises. We were aware of the challenges we would eventually encounter, so when those problems arose, we were prepared.

Great. This may be speculative, but do you think your former company is likely to continue using Acuity in the next 18 to 24 months?

I highly doubt it.

Okay. And if you had to guess, what changes would Acuity need to make in order for your former company to continue using it instead of going the homegrown route?

That's a good question because, let's face it, nobody wants to build their own scheduling system from scratch. It's not an exciting problem to solve. So, if Acuity were to implement group support and offer first-class support for specific healthcare use cases, allowing us to optimize our workflows, that would make a significant difference. And let's not forget about pricing. It would have to be more reasonable, definitely not 50 cents per session, because that's just outrageous. If Acuity made these changes, I believe my former company would be more than happy to stick with Acuity for the long term.