Geoff Charles, VP of Product at Ramp, on Ramp's AI flywheel

TEAM

Jan-Erik Asplund
Co-Founder
jan@sacra.com

DISCLAIMERS

Published on **Jul 27th, 2023**

Geoff Charles, VP of Product at Ramp, on Ramp's AI flywheel

By **Jan-Erik Asplund**

EXPERT INTERVIEW

Geoff Charles

VP of Product
Ramp

Background

Geoff Charles is VP of Product at Ramp. We talked to Geoff to learn more about Ramp's vision for autonomous agents in finance—and to better understand how one of OpenAI's early partners is thinking about AI interfaces in B2B software, iterative improvement of their AI model through a data flywheel, and ensuring trust & safety.

Interview

One foundational technology that enables Ramp is card issuing infrastructure. Another (potentially) is OCR. How do you think about LLMs like GPT-4 as a foundational technology (or not) for Ramp and what does it enable Ramp to build that couldn't exist before?

LLMs are extremely powerful at understanding and interacting with language, and a lot of what we do at Ramp is just that. A receipt, a transaction, an invoice: these are all types of language around a financial event, and our job is to help

finance teams understand, audit, classify, control, and reconcile these financial events.

Before GPT-4, we would have to dedicate a significant amount of resources to build in house models that try to predict what this language means, usually using more classical data science approaches that perform well when there is a structured outcome (e.g. compute the probability of default). But for predictions where we don't have enough historical data (e.g. tell me who this vendor is), using GPT-4 is a massive accelerant.

Ramp recently announced a suite of GPT-4 powered services across expense management, vendor management and bookkeeping. Can you give us context on how this initiative came about? What's the core problem statement around incorporating AI into Ramp and what do customers want that AI can solve?

First, Ramp's mission is to save companies time and money by automating the operations behind finance, starting with expenses, invoice payments, vendor management, card transactions, and accounting reconciliation. A lot of this operational work is slowed down by manual tasks: anything from parsing information from an invoice, to auditing a receipt, to classifying a transaction. We've always leveraged AI to help out customers, now we're able to do even more with GPT-4.

Ultimately, customers want their software to do more of their work for them, so they can focus on more strategic initiatives. They don't care what we use. AI is all under the hood. Your spend management software should tell you if an employee is abusing the expense policy. It should tell you how much you spent last month on the team offsites—without you needing to classify every transaction. It should tell you if you are overpaying for software.

Finance teams just want time back, and they don't want to have to work for software to make this happen. They want their software to work for them. That's what we are doing with AI.

From Ramp's perspective, what did the world look like before GPT-4 and what does it look like after?

Ultimately, GPT unlocks the ability for any company to tap into advanced data science techniques focused on the

understanding of language—a capability that would have taken a significant amount of research that only larger companies would be able to afford. This will accelerate the value that any company can provide to their customers, evening the playing field.

More specifically when it comes to Ramp, I'll use the jobs-to-be-done framework, which asks, what are you trying to do? What, at the end of the day, is the value that you're providing from your work?

At Ramp, our job is to automate financial operations, which means: (1) control an event before it happens, then (2) interpret the event after it happens. And the interpretation can be, what was this? What happened? How do I review and approve this event? How do I classify this event for my financial records?

For example, let's say you're trying to buy software. You start with a purchase order to get budget, then you start a contract, get an invoice, initiate the payment, move the money, account for the actual transaction, and then have to reconcile and track of all these things across several systems.

Before Ramp, finance teams just had access to financial events focused on the payment. They only saw the world in credits and debits. But that's not enough to truly manage spend. Accountants used to be limited to operational teams that make sure that, broadly speaking, your books are accurate—but they're not in the loop in terms of everything else. And they can't actually be strategic partners.

To understand everything that happened beforehand, you need to have a deep understanding of what this thing was: what is a contract? What is an invoice? This was a very manual process before LLMs.

It's the same thing when an employee submits a receipt for an expense: I owe you, you owe me, and what are all these things that happen in between?

Traditional data science techniques were terrible at the interpretation of unstructured data. We got really, really good at big data, but big data realistically with our data science techniques were about the predictability of applying machine learning models to quantitative outcomes.

The perfect example of that is modeling the probability of default. Every lender today has deployed a ton of data science techniques around probability of default. That's how they do their underwriting, that's how they do their credit limits, and that's how they do their capital markets tapes.

That approach largely consists of: how do I apply a supervised or even unsupervised machine learning model on features that predict an outcome that I have a structured solution set for? I have my historical defaults. I probably buy a bunch of default data from credit bureaus. And I use those structured data sets to build all my cool little widgets.

But with a receipt, you don't have that data. You simply don't know what the answer is. You also don't have structured data to build a model in the first place because everything is extremely unstructured.

We looked at companies like Scale, Veryfi and Ocrolus that have OCR models and a ton of mechanical turks spinning out these data sets and it's very, very, very expensive. Then the question becomes: how much am I willing to pay for these services compared to the value they provide?

For us, there's a lot of value because our customers love our product and they spend a lot of money on our products, and so we're able to pay for them.

But now with LLMs, you get a generic version. It's not a model that's built just for invoices that these vendors have, but it's a model that understands concepts and objects that you can deploy and you can get to, within a margin of error, the same amount of accuracy as these vendors at a fraction of the cost.

So now we can deliver incredible value to our customers at a fraction of the cost.

What's an example of a customer paint point that previously could only be solved expensively via mechanical turk that GPT-4 can now solve at a fraction of the cost?

Here are three:

1. Expense intelligence. One of the features that we're actively developing is deeper auditability of expenses. For example—one of the biggest abuses that happens is hotel minibars. It's actually extremely hard to audit hotel minibar usage because it's all part of the same receipt as the hotel, so it tends to slip through unnoticed.

In order to build a feature that enables you to detect discrepancies in a hotel charge related specifically to the minibar, you'd have to build a ton of functionality. Classifying every type of liquor in the world would have taken a significant amount of time. And OCR vendors don't have that data, because they've never needed it. Now, though, you can just ask GPT whether a receipt has a minibar charge, and without any historical data or calibration, it'll be able to identify that.

2. Contract intelligence. Contracts have pages and pages and pages and pages, and sometimes you'll have an important aspect like a termination clause hidden on the 15th page. Now, you can ask GPT, "Hey, what period do I need to say that I'm not renewing? And which email address do I need to use to send that notification? What do I need to say?"

Our vendor management tool will now be able to ingest a contract and, a week before termination is due, send the customer a notification from Ramp saying, "Hey, we've parsed your contract. Your terms are up, and there's a termination clause that ends in 14 days. Do you want to send us this email? It's already preformatted to cancel your contract."

3. Customer service. Even intuitive software like Ramp leads customers to have questions that are not straightforward to answer, but thanks to LLMs, they can just ask Ramp and we can immediately serve them—whether it's an action they want to take, a question they want answered, or an issue they want resolved. This radically decreases our SLAs and increases our operational efficiency.

For more examples, check out https://ramp.com/intelligence.

In finance, many solutions exist as services rather than as software. For example, a Series A startup might have a controller, a fractional CFO and a software-enabled bookkeeping service like Pilot. What do you make of AI as the promise of replacing high cost services with low cost

software, how far does that extend and what do you think some breakpoints will be? What does the finance team of the future look like given what you expect that AI will be able to accomplish?

It's still early for AI to replace high-cost services, mainly because you need a highly trained operator to validate the outcomes of AI. A bad prompt is all it takes for AI to mess up.

What is more likely to happen in the near term is the elimination of low-cost services that high-cost services use, like mechanical turk based services.

AI is not yet able to identify net new strategies to save on corporate taxes, but AI is very capable of identifying how you might want to classify some of your expenses to optimize on existing strategies, and for many companies, that is enough.

What this means is that the cost of these services will go down, and more of the margins will flow from service based companies to those that can uniquely deliver value using AI.

The finance team of the future will be much more strategic than operational. You will need more education and expertise to provide value on top of an AI assistant than your classic CPA, so many degrees will evolve towards incorporating how to use AI, just like we incorporated how to use calculators and later computers in science.

The chatbot or copilot has become the main embodiment of AI from a product perspective. Ramp has both taken the approach of integrating AI throughout the product not only embodying it in a single product feature. It also has a copilot feature. Can you talk about Ramp's approach to interfaces into AI and how it sees chatbots specifically? Is integrating natural language interfaces into, e.g., its Slack integration important to making financial data more accessible?

One of key principles at Ramp is to focus on outcomes, not interfaces.

Many AI implementations are a thin layer on top of a product that customers have to go out of their way to use. But AI should be more than a flashy chatbot interface: it should be embedded in your workflows to actually get things done.

For example, when you're analyzing contracts on Ramp's vendor management platform, you don't want to ask a chatbot question after question—you want the key details extracted and analyzed for you.

Having AI pull out and highlight the most important terms from an automatically imported contract is far more useful than a chatbot that requires manually uploading documents and posing individual, one-off questions. AI should work for people, not the other way around.

How does the interface of software evolve with an extremely strong AI in the background?

When you look at classical software, it's basically forms, dropdowns and buttons. The innovation until now is to digitize the workflows that you've had previously in paper.

A perfect example is Expensify: you come in, drop your receipt, fill out the forms, and submit. It goes to the person who has to approve it, they're prompted to make sure everything is correct, and they sign off. There is nothing truly innovative with this approach.

Our goal with Ramp and AI is to decrease the time you spend in software. This is counterintuitive because more tech companies have engagement goals. Our goal is automation, not session count.

For example, the tool would basically be able to tell you what expenses you owe and would have already pre-classified all of them. You get a prompt when you come back from your trip saying, "Hey, just confirming it was $165 for these three things. There's one thing that I'm not super sure of. Can you just confirm that for us by clicking one or two?" You're done.

As a manager, the experience is like, "Hey, we've audited your 100 employees' expenses. There's two here that we think are worth a conversation. Click here for us to take action?"

For the finance team, it's saying, "Hey, you have 500 employees, you have 99% compliance. There is probably $10,000 on the table. I would recommend changing your policies here, here, and here." And our software will constantly learn and tweak the expense policies accordingly.

It's similar to when pilots went from needing to pay close attention to everything going with their plane to sitting back, with autopilot doing its thing. All they need to do is be notified if anything goes super wrong so they can take control.

Similarly, I think all software is going to become radically more simple and more delightful, and people will spend a lot less time on the operational side of the software and a lot more time on the strategic side of the software. Then, instead of thinking about following up with people and getting their receipts, finance teams can be thinking about the implications of the policies of their company, how to create more cash flow with vendors, how to pre-negotiate their rates, how to reduce their taxes, and so on.

Can you talk about your vision for autonomous agents in finance and what that might look like inside of Ramp? Let's say Ramp gives you an AI controller who autonomously works with a bookkeeper to close the books every month. What needs to happen for this vision to become realized?

Just as we pride ourselves on doing more with less, we want our customers to do the same.

With Ramp, you won't have to hire a large team of financial operators manually reviewing receipts and contracts and entering data manually in stale spreadsheets. Everything is automated and consolidated. In a sense, Ramp is delivering autonomous agents that are doing low level work, so that our customers can spend more time on higher level work.

These agents will be very focused on a key job (e.g. expenses, fraud, accounting classification, contract negotiation, etc.), and will leverage our internal data as well as broader context of language from GPT and the web. This is already well underway.

By using Ramp cards or Ramp's Gmail extension, Ramp aggregates a massive amount of data about customer spending that enables it to save customers money, incentivizing customers to give Ramp more access to data. How do you think about where AI sits in this flywheel and what it accelerates? Does the problem reduce to aggregating proprietary customer data assuming that AI

deployment will get commoditized and become undifferentiated?

AI commoditizes the interpretation of data, not the access to that data in the first place. Few companies have access to the data that Ramp has on what companies are purchasing, how much they are paying, and their specific contract terms. This data typically lives in emails, Google Drives, or for larger companies, in contract management systems. That's our competitive advantage. The more data we get from companies using Ramp to facilitate purchasing decisions, the more we can generate insights to help companies save money, the more companies will use Ramp to make purchases—and the flywheel keeps going.

Vendor management appears to be a place where AI has the potential to turn services margins into software margins. Can you talk about how vendor management, particularly negotiations, worked pre-AI and how you envision it will work post? How can GPT-4 deliver on the promise of negotiation with an automated software approach vs a people driven and manual one?

Now that we can deeply understand contracts, we are able to enrich a vendor payment with context, such as the unit price, pricing mechanics (e.g. price per seat), the contract periods, the renewal periods, etc.

We can instantly benchmark this data to identify discrepancies in price across our 15,000+ customers. We can offer pricing transparency to all customers on our platform for free. We can give them services to help them negotiate, and we can automate most of the negotiations themselves by crafting accurate and personalized emails to vendors that will lead to lower prices.

Before Ramp, companies either had to front the cost themselves with payroll, asking their managers to do this work (without any training), hiring procurement teams that would slow down the purchase process, or they simply didn't bother, leaving a ton of money on the table. Now, we can offer them these services at a fraction of the cost because it costs us significantly less to offer it. We can leverage our expert in house teams, contract data we already possess, and GPT to automate. It's a win all around.

Hallucinations and trust & safety are two major issues for LLM-powered products and nondeterministic output, especially for the highly sensitive use case of B2B finance. How has Ramp built on top of GPT-4 to deal with these issues? What has Ramp done to ensure a high quality, reliable, consistent experience for customers?

Explainability isn't enough—people need control.

The best AI is useless without trust from users. Explainability, which attempts to trace how models make their decisions, seeks to build trust. But explanations aren't always helpful or even possible. It's more important that models improve and are responsive to user feedback.

For example, if Ramp's spend intelligence model incorrectly routes a purchase to the wrong spend program, a lengthy explanation of why it was wrong isn't particularly useful. We simply allow customers to provide feedback so the model learns for the future. Focusing on control and continuous improvement is more meaningful than attempting to explain every AI decision.

Ramp has been moving into enterprise. What do Fortune 500 or 50 companies think about AI-powered products, what are their concerns re sending data to OpenAI, and how does that change what you build?

We take data security very seriously, which is why our contracts with OpenAI and Anthropic ensure that we have full control of the data we share with them and their retention policies, and that their models would never be built on Ramp data. We partner with the most modern companies on the planet, and they are extremely excited about our work in helping them continue to automate.

Additionally, we safeguard data by splitting models into two categories:

1. General models trained on aggregated customer data to handle common tasks. These shared models learn patterns across customers without retaining private data.

2. Sensitive models that temporarily use private customer data without storing it, using a technique called in-context learning.

Customers must explicitly opt-in to share data for these models.
This framework ensures that no customer's information is used without permission. It allows us to build models that understand both general knowledge and customer specifics, without risking accidental exposure of private data.

Can you help us understand the underlying architecture of the AI service that powers the different services in the Ramp product? What reinforcement learning if any that happens? Any key vendors in the "AI stack"?

We can go into more technical detail, but here's the high level picture of how we use AI to enable document understanding and automation.

1. We need to recognize what's important in a source document, and get the data out in a more structured format than raw text streams. For instance, LLMs need to understand how tables are structured in terms of columns and rows in order to extract useful information. We typically use LayoutLM on Azure Form Recognizer to power this.

2. We need to force LLMs to output structured data, including the fields we want (and referencing world knowledge, like Meta being another name for Facebook) without hallucinating irrelevant things. We typically use GPT-4 or Claude on top of Rahul's (Head of Applied AI Platform) Jsonformer project to power this.

3. We need to perform semantic search over documents, to find specific pieces of data or correlate documents with each other (i.e. finding contracts from the same vendor to power pricing intelligence, or identifying approval patterns to find auto approval rules). This is done using a combination of local embedding models and Amazon's OpenSearch database.

Ramp long has sought out and teamed up with best in class partners, e.g., Stripe. Ramp has partnered with OpenAI, Ramp's a customer of OpenAI's and OpenAI is also a customer of Ramp's. Can you talk about Ramp's partnership with OpenAI?

Ramp co-founder and CTO Karim Atiyeh's biggest focus is identifying emerging talent and emerging partners. He's always

basically planting the seed for something that'll pan out in 3, 5, 10 years.

For example, early on, we were super early adopters of no-code solutions like Retool. I think we were one of their first customers. Similarly with project management software like Linear—we went very, very early on these guys.

Marqeta was an early partner. Obviously when Stripe started doing Stripe Issuing, we partnered very, very early with them as well and leveraged Stripe versus Marqeta to continue improving both sides.

Same thing with emerging talent—we're always trying to find the talent that is early so that they need you a lot and they're listening to you and then become very basically an extension of your dev team. That's kind of been the strategy so far.

When it comes to OpenAI, we're obviously close to them and Satya Nadella who's also a Ramp advisor. We were early adopters of the OpenAI technology to build new capabilities for our customers. So there's a lot of good synergies between two companies.

Tell us about Ramp's acquisition of Cohere.io and how it fits into Ramp's AI strategy.

On the Cohere team, Yunyu Lin was an early engineer at Ramp who left to start his own company. They focused on customer support, but over time, they became more of an AI company, after seeing that a lot of the value that they could provide to customer support teams and companies through the application of LLMs on top of conversations with customers.

When you think about what Ramp does, Ramp provides context on top of conversations with employees around financial events. There's actually a lot of synergy between what they've been able to do with text-based support tools and what we've been able to do with transactional data.

With OpenAI and the step function change in AI, Karim concluded that we needed more firepower internally to understand this technology and leverage it both for internal purposes and for customer and product purposes. He reached back out and found a deal to make it work.

Can you talk about whether Ramp evaluated other LLMs like Claude (Anthropic), and how Ramp thinks about designing for optionality versus building specifically for one LLM?

We actually use both GPT-4 and Claude internally for different use cases, as well as a variety of local fine-tuned models. For us the biggest consideration is UX—GPT-4 has great quality but is very slow and expensive, Claude is somewhere in the middle, and local models are very fast and cheap but work well on simpler tasks. Our framework here is—where in the product does speed matter vs. quality? Is the interaction realtime, or is it asynchronous where the user can afford to wait? Is the task intrinsically difficult (e.g. understanding contract terms) or easy (classifying transactions)?

On optionality: distilling third-party models into single-task local models is very easy and we've done it in the past. We trust our business partners to be good partners, but given the above we view the business risks here as essentially nonexistent.

Ramp has emphasized speed in product development as its major competitive edge. How has GPT-4 helped Ramp build faster, ie as a generic API for turning unstructured data or semi-structured data into structured data?

GPT-4 is not by itself a competitive advantage as a technology (just like having a computer is not a competitive advantage - everyone has one). It's about how quickly you are able to adopt it and train your teams on how best to use this technology. We were one of the first companies with a contract with OpenAI and an integrated team that was training everyone at Ramp on how to use it to be more efficient with just about any task.

Ramp has 500 employees, Brex has ~1,000 and Rippling has ~2,000. How has AI changed how Ramp hires and builds its team internally, and how it thinks about doing more with less?

We can continue to do more with less. For example, we can build chatbots that deflect a ton of support tickets so we can continue scaling with a lean support team. We can write code faster. We can deploy value added services faster. We can do

customer research faster (we built a bot that scrapes all our sales calls and gives us insight on what our customers need). The goal is to stay as small as we can so we decrease the overhead of coordination, and continue giving each individual employee a massive, motivating scope.

"Software is eating the world" has become "AI is eating the world". How does AI become an accelerant to ambitious teams going multiproduct and eating up adjacent use cases? How does AI change the trajectory of company building?

Let's start from the ground at the data layer. At Ramp, we had a homegrown system—what we called merchantilization—where you swipe your card, and it's a Square merchant, but actually you'd know that it was, for example, Blue Bottle. That was fairly complex. We had an in-house model. We switched to an LLM, which has been way better—exponentially better.

Relatedly, the pain between transactional data that's card and invoice data is massive. That's why we do what we call "vendor unification" across those two things, and we've recently launched a new version of Ramp Vendor Management which does that.

Vendor unification helps finance teams understand that actually, say, Amplitude is the same thing—that people might be paying for it on the AP side and also paying with their corporate cards, and that you could put more things on corporate cards, which helps us as well.

In terms of company building and the trajectory that a product takes over time, I think what's happening is that in the past, companies started with point solutions and then they would add a bunch of features or products to become consolidated platforms. Most companies go through that motion.

The issue is that the companies that have made it to the other side of that have lost their ability to build delightful software because they become huge and sales-driven, and their talent has largely evaporated.

For example, a Coupa or Workday, was pretty good back when they were founded. They had really good talent and they built some really great products, and then everyone had a financial

event or a financial exit, and then they kept selling and selling and building and building, and then now, people look at Workday and they're just like, "I hate this experience" So then they start all over again by signing up with Ramp, right? That's my hypothesis. People had to make the trade off between delightful point solutions and robust but poorly designed consolidated solutions. The industry went back to point solutions for a little bit. Now, there's these new entry points for point solutions that are working towards building these larger consolidated platforms.

The question for me is how Ramp or other competitors can continue building a consolidated platform without becoming the thing that we set out to destroy?

That requires a constant focus on user experience and a constant resistance towards sales-driven culture. I'm not saying sales is the worst thing in a company in terms of culture, but at a certain point, you can forget why you were successful in the first place.

What can AI do there? I think it can help you build faster and make your engineering teams more efficient. We just launched an AI agent that summarizes customer calls and does most of our research. We can ask, "Why did people drop off? Why are most customers angry on support channels? Why did this person choose a competitor versus us?" AI can do a lot of these things to help you continue to maintain truth around what truly matters.

Then, when it comes to the complexity of building multiple products on top of a single platform, hopefully AI can facilitate a lot of that through lowering the cost of development, but we're still early.

Disclaimers