

EPISODE 734

[INTRODUCTION]

[00:00:00] JM: Market strategy defines how a company is positioning itself to be successful. This strategy encompasses engineering, sales, marketing, recruiting and everything else within a company. Herb Cunitz has led teams at Hortonworks, VMWare, SpringSource and several other companies over his 30-year career in software. After working as president of Hortonworks, Herb started AccelG2M. AccelG2M works with software companies to define their go-to-market strategy.

Software companies require a great deal of long-term strategic thinking. Engineering, sales, marketing and leadership all must work together to build a plan that will allow the company to reach an exit either an acquisition or an IPO. Executives at a software company must create a clear strategy and communicate it to the employees throughout the organization. The strategy must be implemented, meeting deadline and hitting milestones. New team members must be recruited and unsuccessful workers must be let go.

In today's show, Herb provides some valuable strategic wisdom for anyone working in software, whether you're an engineer, a sales person or an investor. Before we get started, I want to mention we're looking for several positions. You can find job openings at softwareengineeringdaily.com/jobs. We're looking for several journalists, an entrepreneur in residence and several different internships. There are other roles too. You can go to softwareengineeringdaily.com/jobs to find out more and we'd love to get your resume.

[SPONSOR MESSAGE]

[00:01:51] JM: Your audience is most likely global. Your customers are everywhere. They're in different countries speaking different languages. For your product or service to reach these new markets, you'll need a reliable solution to localize your digital content quickly. Transifex is a SaaS based localization and translation platform that easily integrates with your Agile development process.

Your software, your websites, your games, apps, video subtitles and more can all be translated with Transifex. You can use Transifex with in-house translation teams, language service providers. You can even crowd source your translations. If you're a developer who is ready to reach a global audience, check out Transifex. You can visit transifex.com/sedaily and sign up for a free 15-day trial.

With Transifex, source content and translations are automatically synced to a global content repository that's accessible at any time. Translators work on live content within the development cycle, eliminating the need for freezes or batched translations. Whether you are translating a website, a game, a mobile app or even video subtitles, Transifex gives developers the powerful tools needed to manage the software localization process.

Sign up for a free 15 day trial and support Software Engineering Daily by going to transifex.com/sedaily. That's transifex.com/sedaily.

[INTERVIEW]

[00:03:39] JM: Herb Cunitz, you are the principal owner of Accel G2M. Welcome to Software Engineering Daily.

[00:03:44] HC: Thank you, Jeff. Happy to be here today.

[00:03:46] JM: We've done a few previous episodes with your colleagues, Shawn and Mitch, and what I liked about both of those conversations was the high-level strategic discussion. Your company is AccelG2M and you work with CEOs and the board and other parts of the company to define high-level strategy. Tell me the story of how you started AccelG2M.

[00:04:15] HC: Sure. All of us, Shawn, Mitch and I, had worked together in various companies, enterprise software companies, open source companies, etc., all early stage and going through what I would call the traditional phases of growth, and those would be if I'm a very early stage company, how do I get to a minimal viable product. If I'm the next stage, how do I get my first set of beta customers and how do I get my first set of paying customers, then how do I build a

repeatable sales model around that and growth and product model and then how do I start scaling. That right there takes a company from call it zero to 100 million.

What we saw as we worked through it at many of the companies we were part of, is that companies can start to slow down or fail at different stages in that growth curve. The most common one is I call the trough of despair, which is how do you get from that first set of customers that are paying you to a reputable sales model? That usually happens somewhere around 5 to 10 million in sales.

What we realized is AccelG2M, we've been through that multiple times have seen what it's like to have the right business model in place and felt that we could help the CEOs of software companies who are going through that same growth curve, and they tend to be technology founders with fantastic technologies that they're trying to understand, "How do I best build out my sales and go-to-market strategy and team and execution to get through that trough and get to the other side?" We found that's a fun area to work with different companies out in the market.

[00:05:55] JM: Why does that failure tend to happen when companies get to 5 to 10 million in sales?

[00:06:02] HC: Usually it happens in a couple of ways. The most common is they don't have a reputable sales model yet, and what that means is the product that they've built does not have the right product market fit. It doesn't mean it doesn't exist. It means they haven't founded yet to say who is the buyer type of company? What is the persona of the buyer? What problem are they trying to solve and it is consistent across a large set of companies? If it is consistent, then they're able to go build a model around it. How many sales people do I need? What other product features do I need? How many SCs do I need and how do I go grow this?

Where the challenge comes in is where it's not repeatable yet. They may have 10 or 20 customers and all of them are using the product for different reasons. That's not repeatable yet. That's 10 great one-off customers. To make it repeatable, you need another 10, 20, 30 who are using it to solve the same problem.

[00:06:58] JM: What's the process of honing in on the use case that is representative of a larger market? If you have — if you go to that situation where your company is selling a product or selling five different flavors of the same product because you're just trying to figure out what exactly your product market fit is. What's the process of narrowing down the piece of software that you should actually be selling and defining a repeatable sales process around?

[00:07:29] HC: Great question. So first let me answer that, by the way, having 5, 10, 15 customers all using your product for different reasons in the early stage is normal and good. So that's going to happen. This is not something you're going to avoid. It's can you make it repeatable after a period of time? The process for figuring that is, first, just look in the mirror and you have to be honest to say, "What problem does your product solve?" Not, "What is your product do?" That's easy, and every technology founder can, in great detail, articulate what their product accomplishes and what it does and what features it has.

The other side of the question is who cares about that? How do they use it and what problem are they using your product to solve? As you really start to define that in concrete terms of, "We're specifically solving this problem. Here is the series of companies that have that problem. Here's what their alternatives are and here's how we solve it better than those alternatives." It's running through that process probably by talking to those customers asking them why they're really using the software. Why did they buy it, and then looking for those patterns, connecting the dots where you can see those patterns. In many cases, the actual use case, the thing that they're using it for, is much simpler and more mundane than the product was designed for, but that's okay. That gives you an initial point to start growing the business and expanding from that.

[00:09:08] JM: It's worth pausing here and exploring why this information, this topic, is useful to the software engineers that are listening to the show, because I think most of the people that are listening are working as an engineer. Maybe they're a lower-tier, a mid-tier engineer at a company and they're thinking, "Why do I care about this? Why do I care about product strategy? Why do I care about enterprise product strategy?" Why is that useful for software engineers throughout an organization to know?

[00:09:37] HC: It is very useful for a couple of reasons. First, if I'm a software engineer building a great technology, I'm clearly invested in that technology and that company for both my career

and my personal life and just what I can do with my technology prowess. First part of it as they're thinking about that, they want to be a part of a successful enterprise. If ultimately the enterprise does not figure out how to monetize at scale, then they probably won't get their next funding round, and that is one area where many of the companies we've worked with as we've talked to the engineers, the whole funding side of the business moving from series A, to series B, to series C seems to be somewhat of a black box. They're trying to understand, "Well, why couldn't we get funding? Why didn't companies just want to throw money at us?" It comes to when you're moving from, say, series A to series B, you have to show that you have an understanding of how to monetize the business. That's clearly important and it should be important to the engineer. That's one core area of it.

The second is, at some point, many software engineers would like to see some return on investment, and if their investment is the time and energy they're putting into that company, they'd like to see a return either in the form of public offering or an M&A where it's two companies coming together where they can realize the dream of that technology going to a much broader market. All of that comes down to better understanding how to monetize the business and getting through that first chasm of figuring out how to build a repeatable sales model.

I give the last piece, which is when companies are functioning well and they are growing, engineering and sales is a good way to think of it as a great matching of supply and demand. The supply side is engineering, building a certain set of products and capabilities for the market and taking that out to market. The sales side is how do you build a distribution engine I can take that supply, those products, and bring them into the market. In a well-matched company, supply grows demand. What the engineers build is easily brought by the sales team to the market. When you start having a mismatch of too many products that don't fit well in the market, then you're loading up products that don't get sold into the market and people get frustrated. If you have too much supplies or too much demand, too much on the sales side and not enough product, suddenly the sales team is off selling ahead of what the capabilities are, which is not great for the product team, but when it's well-matched, you've matched that supply and demand and now you've got a good bidirectional interaction between the market.

In this case, usually the product team becomes the interaction between sales and SEs back to the engineering team. So you get your requirements from the market filtered back to engineering and you get good product out to the market. Having that work in a harmonious balance is incredibly important as a company scales and something important to all engineers over time.

[00:12:44] JM: With this variety of companies that you've worked with at AccelG2M offering them some advice and helping them paint a strategy for how they're going to market, can you give an example to help make it concrete for the listeners what exactly you do an example of a problem that you have helped a company solve?

[00:13:10] HC: Sure. So I won't use specific client names here, but more conceptually what were the problems they were trying to solve. So very common one, and this is actually very common across many of our clients, is let's assume you're probably a \$10 million company. You have some repeatability in your sales model.

[00:13:32] JM: By the way, sorry, \$10 million company. Does that mean a \$10 million valuation, or \$10 million in sales? What does that mean exactly?

[00:13:39] HC: Great question. \$10 million in sales, so billings. So \$10 million in annual billings.

[00:13:45] JM: Got it.

[00:13:45] HC: So if somebody were at a \$10 million in annual billings and are starting to grow the company, a very common model now is companies have moved towards a SaaS model or a cloud-based model. They've tried to build a distribution engine or a sales and marketing team around inbound. What that means is more of a lower-cost sales team that takes a lot of the inbound interest through the website for the product and converts that over the phone to a customer, and usually at an average selling price, let's call it below \$100,000.

That's a very common model. The venture capitalists love that. It's a lower cost of sales. You can scale the company quickly. Many companies have done that. A very common issue is now they're trying to penetrate the large enterprises, the very large companies, large financial

companies, like J.P. Morgan, Morgan Stanley, Credit Suisse, the large telecommunication companies, AT&T, Verizon, Sprint, etc. As they're trying to penetrate those companies, those companies don't work with just traditional, "I sent you an inbound interest over your website. Can you just sell me something over the phone?" They require a high-touch. They require probably something customized and they require some services built around it. That requires more of an enterprise sales to a more traditional enterprise sales team.

So common a common and that many companies are facing now is how do I complement my inside selling model, which is great and low-cost and low-friction with those more expensive enterprise sales teams, and can I complement the two together to go after a broader market opportunity? That is – And helping to put that in place and adjust the culture of the company to be able to successfully bring in that type of team and make them successful in the market. That would be a very common inflection point that we're helping companies through right now.

[00:15:45] JM: So in that process of developing that repeatable sales model where you're trying to go from, for example – Let's say you've developed 10 million in sales to startups. There's a bunch of startups that need your product and you're starting to exhaust the startup market. So you're starting to say, "Okay, let's go sell to Verizon, or to Credit Suisse, or to Bank of America." What kinds of things do you have to build? What you have to do in order to sell to the banks, or the hospitals, or the really large companies that are also, by the way, going to give you much larger returns on those contracts?

[00:16:27] HC: Correct. So those larger companies typically will have a larger return, meaning potentially in the initial sell. But for sure the lifetime value of that company to you as a customer is worth much more than a smaller ad tech startup or someone else you may have been selling to. If you're going to start talking to them, hopefully you don't need much change in the product.

Now, many of those companies will be demanding and ask for additional features or capabilities. So you may have to be flexible on adjusting your product strategy to bring in some of their specific features ahead of where you may have wanted to, and that's common. That would be one area they'd have to think about.

Second is in many cases, those larger companies have more integration requirements. I need your product to work with my existing analytics tool, existing databases, existing cloud provider, etc. So you may have to do more integration capabilities than your thinking about, APIs, than you were thinking about your initial product strategy.

Third is you'll probably need some level of training and services, because these larger companies want to use it across a very broad team, maybe tens, hundreds, thousands of people, and that presents a scale issue where you may have to train or enable them to be successful, require better documentation, or build more retraining and services capability to make them successful. Then you have, we talked about earlier, the whole sales and marketing engine is how you reach those companies and how you would interact with them from a sales and marketing perspective. But from a pure engineer's perspective, new capabilities, new integration capabilities, deeper documentation would all be areas that they'd probably be called on to deliver as you start moving into those larger enterprises.

[SPONSOR MESSAGE]

[00:18:29] JM: A thank you to our sponsor, Datadog, a cloud monitoring platform bringing full visibility to dynamic infrastructure and applications. Create beautiful dashboards. Set powerful machine learning based alerts and collaborate with your team to resolve performance issues. You can start a free trial today and get a free t-shirt from Datadog by going to softwareengineeringdaily.com/datadog.

Datadog integrates seamlessly with more than 200 technologies, including Google Cloud Platform, AWS, Docker, PagerDuty and Slack. With fast installation and setup, plus APIs and open-source libraries for custom instrumentation, Datadog makes it easy for teams to monitor every layer of their stack in one place. But don't take our word for it, you can start a free trial today and Datadog will send you a free t-shirt. Visit softwareengineeringdaily.com/datadog to get started.

Thank you to Datadog.

[INTERVIEW CONTINUED]

[00:19:37] JM: One of the companies that you worked was Hortonworks. You are the president of Hortonworks, and this is a large Hadoop vendor, or that's how Hortonworks originally was positioned. Now it's more of a broad data services. The customers of a large Hadoop vendor can be quite large. It is around – I mean, it sounds like you had to solve this problem of going to market from the ad tech companies that are just startups and they're more tech savvy and they can kind of figure out your product. You had to go from that to the banks and the telecoms and the manufacturers that do want to do a Hadoop, but they don't know what a Hadoop is. They don't know how to get the Hadoop. They want the big data. They want the data that is the oil. How did the negotiations work when you were developing relationships with those large clients? Do you have any tips around negotiating with really large enterprise customers?

[00:20:37] HC: Yes. So let me preface the question with a couple things around the earlier days at Hortonworks, what you described. I was president and worked with Mitch and Sean and many others clearly as we grew the business. When the company was started, Hadoop is a market already existed. Others like Cloudera and even MapR were out ahead of us, like probably three years, and it pioneered working with some of the larger companies out there.

So some of that motion was already in place and the path was a little bit paved. But when we came in since we were late, we felt we had a number of things we had to do. One was clearly establish a brand in the position, and we did that around pure open source and what that meant to be enterprise, and I'll come back to that for the large enterprises.

Second thing we realized is because we were late, we had to invest very early and very heavy in sales and marketing to go push out quickly into the market. Then the third thought we had was we would need to go enable the ecosystem, and by that I mean if you're in the ecosystem, you're a Microsoft, a Teradata, a business analytics provider, how do we make sure that they knew about us and they worked well with our software? So when we walked into a large enterprise, the Verizons, the Credit Suisses and others you mentioned, that we could say, "Yes, we've worked with this particular instances, call it Microsoft, or Teradata's data warehouse," and it is certified or it works well with our Hadoop distribution. That took a lot of friction away from working with the larger enterprises.

Then the last thing I would say is we made a conscious focus that in the very beginning we were going to go after only the large enterprises with the belief that the lifetime value of a customer would be larger in those companies versus smaller companies. So we avoided and, effectively in some ways, ignored the much smaller company market for the first couple of years of our life. Once we'd established that great position in the Fortune 1000, then we went back and opened up a team to go after the smaller companies.

With all that in mind, how do you then negotiate as a small company with those larger enterprises? One is you have to be confident that you deliver a great value proposition to what problem they're trying to solve. Again, it's less about the technology, but you have to think about what problem are we solving? How would they solved it without us? What would it cost them to solve it without us? How do we align ourselves against that value?

If you understand that well, then from a negotiation perspective you're actually saving them money compared to their alternatives. So it's, yes, you're negotiating for pricing and all of that, but you're helping them save money against alternatives.

The second part that is common is many larger companies, especially when you're small, will say, "Hey, can you just give me three years unlimited distribution? All of your product at this particular low price and I can deploy as much as I want." In early days, that sounds great, but the reality is if your product is successful at enterprise and it has a viral usage and spreads, you've probably sold yourself short by many millions of dollars.

So it's much better if you can come up with a way of doing annual agreements that you reset the threshold based on pricing, based on the volume of usage, versus sign up for a very large multiyear that feels good upfront, but in the end you've set yourself up to probably not even get close to the value that you deserve based on what you're helping that large enterprise with.

[00:24:22] JM: What you're saying there, when you establish an annual contract and then a year later you come back and you say, "Hey, we want more money, because we're helping you more." Is that ever an uncomfortable conversation when you kind of have to tick up the price with the pre-existing customer?

[00:24:41] HC: It is almost always uncomfortable. Those companies have very experienced procurement teams who know easily how to go work with companies from the largest, like Oracle, to the smallest, say, Hortonworks, in their early days and assume they are way more experienced than the person on our side of the fence. So, yes, it's uncomfortable, but I would say you're not actually raising their price. Their unit price probably should go down because they're using more volume. Their overall cost will go up on an aggregate basis just because their volume has gone up. So you're giving them better unit economics. You're just increasing their total spend, because they're using your product more extensively. That's a fair bargain on both sides.

[00:25:30] JM: I see. So you might say even when you're establishing the first year of a contract, you say, "Hey, you're going to get this many MapReduce jobs, or this much terabytes in HDFS storage, and this is the unit economics." Then when year two comes around and they're renewing, you can say, "Okay, now you get a slightly better price, but because you're using more, we're just going to charge you more," and I guess that reduces the difficulty of the conversation and it at least gives you some numbers to be anchored to make a sane negotiation.

[00:26:08] HC: Yes. In your first year, what would probably do is set up a volume procurement schedule that would say, "From 0 to 100 servers, your cost is 2,000 per server. From 100 to 500, its 1,800 per server, and from 500 to 1,000, it's 1,500 per server." It's well-defined for them as they know as they grow what their cost per unit is.

When you now get to your second year, even though – And I'll say this from the company's side, you've laden that procurement schedule, you've had a great discussion upfront, you've set it all and a year from now they're using thousands of your servers, which is a great problem to have. You just should be aware that whatever you put in that initial negotiation initial contract becomes the starting point for that large enterprise to go negotiate with you. As much as you may have thought that was the endpoint, that becomes their starting point.

[00:27:05] JM: Right. You're an expert in software sales. You've been doing this for a pretty long time, and there are some situations in software where your company has completely dominant pricing power. So Microsoft, for example, in the 90s, it was dominant. It's the only

place you can get – I don't know, Microsoft SQL Server or something. I'm not sure what software. So pricing strategy is going to, in some ways, hinge on how dominant your solution.

Like today, cloud providers, I guess there's a little bit of fungibility there. They're a little bit interchangeable. Not entirely. I mean, mostly not entirely, and then like the Hadoop providers, there's some interchangeability between them. So you don't have as much pricing dominance there. How does the dominance of your product, how does that affect the pricing strategy when you have that dominant lead?

[00:28:08] HC: So it certainly could affect it, and if you have a dominant position, you could leverage your pricing strategy in some ways, say, against customers. Personally, that's a great long-term strategy. I think you're better off figuring out a couple things. One is my belief is it's always better to get a customer early than to get it at the highest possible price. By that I mean think of Hadoop and HDFS as you were describing, it's really managing data. Data is very sticky. You're better off getting that customer early getting them started, making them successful at a reasonable price point. If you're doing your job right, they will grow usage over time and you'll help them, and as you develop or acquire additional products, you will have new things to go sell them that will increase the overall cost they have with you. Again, it may not be increase in the individual price per product.

So there's – And that's a better way to grow. Now, does that customer always have the leverage of saying, "Hey, your Hortonworks, I'll switch to Cloudera." Certainly, they do. In reality, very few companies want exercise that right, because it's expensive. While they may save on your product, the other 10 products that it integrates with that go along with it may have to get changed, adapted, updated service, etc.

They tend not to want to do that even though they will use that as leverage. I'd say the more important thing is get the customer early, make them successful. I mean, truly make them successful. More volume will come out of that and then figure out what are other tangential spaces or products that you can build and deliver to them to find other avenues to go monetize the relationship.

[00:29:59] JM: You were at VMWare from 2009 to 2012 after VMWare required SpringSource. For people who are listening that weren't involved in software back around 2009 era, what role did VMWare play within the enterprise and what role does VMWare play today?

[00:30:20] HC: Yeah. If we go back to then, we came in through Springsource's acquisition. SpringSource being a Java development framework. VMWare at the time was clearly doing extremely well multibillion dollar company already focused on the virtualization market. At that time, cloud was – It existed clearly between Amazon and others, but it was pretty nascent still. The public cloud was not clearly as prevalent or as popular as it is today as a place to run your workloads.

At the time with VMWare, their focus, which hasn't changed, was how to they have more workloads stay on a virtualized machine versus moving to the cloud, and in their case more on-premise versus at the time moving to the cloud. The SpringSource acquisition was very much a way of saying, "Can I capture customers earlier when they're developing applications that those developed applications will run better on a virtualized environment, a VMWare VSphere environment?" That was the genesis for the acquisition, a lot of what we worked through, and ultimately where that spun off into Pivotal, which is now a successful software company, Pivotal Software. Has that changed with VMWare today? No. Their core product is still VSphere and many of the things that do run virtualization.

What is different though for VMWare is they now must work in a hybrid world. So they must work with public and private cloud providers and on-premise data centers. The second thing they must work with, and they are doing a good job threading that needle, is the whole movement to containers. As applications are being developed for containers and then orchestrated with Kubernetes and all of the other systems around that, that is one way is an existential threat to a virtualized environment, but it is also a potential opportunity if they can embrace that and allow those containers to run well in their environment, which is the focus that they're doing with some of their recent acquisitions and the things they're doing in their go-to-market strategy.

[00:32:24] JM: Right. They acquired Heptio, which is the Kubernetes company that was started by some of the founders of Kubernetes. If you're VMWare in this situation, you've just acquired a

company that has a core competency in a technology that can disrupt or refurbish your existing customers depending on how you look at it, and it can also open up new markets, what are you doing with that acquisition? Once you've acquired Heptio, and I realize you're not at VMWare anymore. I don't think you have any like super privileged knowledge or maybe things that you've noticed at dinners or something. You probably have some knowledge, some background knowledge, but speaking agnostic of that knowledge, what would you do if you're VMWare with that Heptio acquisition?

[00:33:15] HC: Just for clarity, I have no knowledge. So this is truly opinion. If I'm VMWare and I want to take advantage and help monetize in the Kubernetes space. So on the one hand, the company that have probably more influence in a Kubernetes space is Red Hat and Amazon in terms of the work that they've done there. Clearly, Red Hat now being acquired by IBM, a lot of that is going to be leveraged by IBM in terms of what they're trying to do.

If I'm VMWare, I want to say, "How do I stay relevant in that space? So one way if I have experience and understanding of how Kubernetes work, and that's what Heptio brings them, as you mentioned, some of the key original architects of Kubernetes. So it brings them a deep understanding of how that works. I think it allows them to figure out how do they add value with their virtualized environment, the things other VMWare does in a hybrid cloud in a Kubernetes environment. Then ultimately, how do you go build the other services that Kubernetes will need? Monitoring and management? Service mesh? Orchestration between different containers? How do you start to create some of those other capabilities as VMWare and potentially offer those to the market as a way to make Kubernetes run better in a virtualized environment? Those are some of the ways that [inaudible 00:34:39] if I were there, I would look to leverage the Heptio acquisition.

[00:34:43] JM: And would you start with going to, I guess, your existing VMWare customers and you say, "Hey, we've got this new Kubernetes technology, and we can help you." You focus on the existing customer base, right?

[00:34:59] HC: Yes. You'd focus on the existing customer base who's exploring the usage of Kubernetes and you would try to keep those customers working in your environment versus dispersing into a different environment, a Red Hat environment, an Amazon environment, etc.

This is where you're great. While everyone may compete in that market, they're all partners as well.

[00:35:23] JM: Right. When you were at VMWare from 2009 to 2012, you were in charge of variety of areas of growing the business. I was looking at your LinkedIn, it's like the number of things that you were involved with, it was like growing sales, consulting, presales, and VMWare has tons of clients. It's got tons of opportunities for expansion, and you had all these different areas that were within your purview. How did you decide what to prioritize or what was your process of devising a strategy when you joined this company that had so many opportunities in front of it at the time?

[00:36:04] HC: A couple of aspects to that. So when SpringSource was acquired by VMWare, it was –As I described, help get earlier in the cycle of building applications in the job environment that ran well in a virtualized environment, because you take that as the core thesis and premise. When VMWare acquired SpringSource, there's a lot of ways you can acquire a company. Their choice was to keep us as a separate unit inside of VMWare. Think of us as a wholly contained, almost like a subsidiary inside of VMWare continuing to sell the SpringSource [inaudible 00:36:40] and capabilities to VMWare customers and others.

The company did not get tightly integrated and dispersed through the company engineers, and to the engineering org sales, and to the sales org. We kept in a separate unit. That was a conscious choice, because the customers that we sold to the value proposition and how you reach them was different than VMWare. VMWare had a very and a phenomenal channel model how you could sell through channel and to customers. SpringSource had a very much more direct model, very little channel at the time. So they kept it separate.

So in many ways, the strategy was how do you run it as an entity just like you'd run an independent software company? But how do you leverage the VMWare engine, which at the time was many hundreds or even thousands of salespeople, customers, partners, etc. How do you enable that engine, those customers and partners to know the value of SpringSource and be able to leverage that in their environments? It's much faster to leverage that environment than it is to try to organically, I'll say, scale your own small company? That was the core of the

strategy, and that required services and training and enablement of their team and great salespeople could go out and sell and service these people and SEs, etc.

[00:37:58] JM: It sounds like you found that focus of what to work on from thinking in kind of a top-down fashion. You knew that the way to make the acquisition successful was to focus on how SpringSource and VMWare were additive to one another, and then the tactics all stem from what that high-level strategy looks like.

[00:38:22] JM: Correct. With the ultimate goal of, in this case, how would us as SpringSource go grow the product revenue, but more importantly, how do we make VMWare more successful? How do we help sell more VMWare product? How do we help their core business be more successful? Ultimately, in any acquisition, you have to think through. If you're an acquired company acquiring to another entity, it's now they're accompanying you. How do you help them better sell their product? It's just like if you're small company and you want to partner with another company, a very large company. If you want to partner – If you're a small company and you want to partner with Oracle, or IBM, or VMWare, or Amazon, etc., it doesn't matter what your product does. It's how does your product help them sell more of their product? If you can crack that code, you've got a much stronger partnership. Same in an acquisition.

[00:39:17] JM: Do you have any other broader – You've seen a lot of acquisitions. We've just talked through a couple of them. How do you run a successful acquisition both from the point of view of the acquirer in the acquire?

[00:39:30] HC: There had been many books written on that topic, and I don't know that anyone's completely cracked that code, especially because most would say that most acquisitions don't yield the value that they expected upfront. But I would say what are some things you can do to improve your odds of having a successful acquisition? One is if you're the acquired company, how do you best help the larger entity sell their product? How do you make them more successful?

Second is I do believe the faster you integrate a company in, the better. Meaning, many small technology companies, groups of engineer, salespeople, etc., get acquired, want to or go into this believing that, "I get to keep my small company and I just get to go leverage this whole big

company and life is going to be wonderful.” It doesn't really work that way. You're now part of that bigger company. You have to follow their rules, their processes, their go-to-market, their way that they build and deliver software, their way that they determine which software products get funding and more engineers to go build.

So the faster you can learn their model and fit in, the better, versus trying to stay separate and, I'll say, retain the culture and the operating model that you had as you went into the entity. If you do that right, your technology that was just acquired can see a much broader market and can realize the vision that you as a technology founder always had, which is to become much more prevalent in the market. The reason it can is it goes through a much larger sales and distribution force hopefully more effectively. Those would be some of the things I would look at, but I don't think there is a candid formula on how do you make an acquisition successful. They're all different.

[00:41:18] JM: You might've been speaking metaphorically, but are there actual good books that have been written on acquisitions that you know of?

[00:41:25] HC: There are some, and none that come top of mind right now, but there certainly are, I'll say, some case studies and some other things that have gone well in terms of acquisitions and also, obviously, ones that have not gone well. So there certainly are great case studies that go through some of those and it's always easier to look in the rearview mirror, but then dissecting as to went well and what didn't go well.

[00:41:48] JM: Yeah. What is it about that phenomenon, what you alluded to, where a lot of times the acquisition looks like it's going to work out really well, but then in practice it just seems to fall apart or it doesn't work as well. There's not as much leverage as people anticipate. Why is that?

[00:42:04] HC: Probably a couple of reasons you could end up there. One, there was a cultural mismatch between the teams, just the body rejected the organism. So that happens often in an acquisition. That's one way that it may not work out, is if the cultures just can't align. Second reason would be the assumptions on the market opportunity were too rosy. So the assumptions

that were used to justify the acquisition were too rosy compared to what the true potential or where the market is going to go. That would be a second one.

Then if the growth doesn't come as expected, the acquired company gets blamed that it shouldn't have been acquired. It wasn't the right fit, etc. I'd say those are probably the two most common reason, a cultural mismatch or overly optimistic assumptions on what the potential of the combined entity is.

[SPONSOR MESSAGE]

[00:43:10] JM: Managed cloud services save developers time and effort. Why would you build your own logging platform, or CMS, or authentication service yourself when a managed tool or API can solve the problem for you? But how do you find the right services to integrate? How do you learn to stitch them together? How do you manage credentials within your teams or your products?

Manifold makes your life easier by providing a single workflow to organize your services, connect your integrations and share them with your team. You can discover the best services for your projects in the manifold marketplace or bring your own and manage them all in one dashboard. With services covering authentication, messaging, monitoring, CMS and more, Manifold will keep you on the cutting-edge so you can focus on building your project rather than focusing on problems that have already been solved. I'm a fan of Manifold because it pushes the developer to a higher level of abstraction, which I think can be really productive for allowing you to build and leverage your creativity faster.

Once you have the services that you need, you can delivery your configuration to any environment, you can deploy on any cloud, and Manifold is completely free to use. If you head over to manifold.co/sedaily, you will get a coupon code for \$10, which you can use to try out any service on the Manifold marketplace.

Thanks to Manifold for being a sponsor of Software Engineering Daily, and check out manifold.co/sedaily. Get your \$10 credit, shop around, look for cool services that you can use in your next product, or project. There is a lot of stuff there, and \$10 can take you a long way to

trying a lot of different services. Go to manifold.co/sedaily and shop around for tools to be creative.

Thanks again to Manifold.

[INTERVIEW CONTINUED]

[00:45:25] JM: So when it comes to company exits, the two most common forms are – Well, the two most common success cases are acquisition or IPO. You've been involved in three IPOs. I want to ask you some basic question around that. First of all, why should a company IPO?

[00:45:44] HC: Good question, and first to your comment, I've been involved in both IPOs and acquisitions, and given a choice, putting aside everything we just talked of acquisitions, I think an acquisition is probably the better and a fantastic exit option for a young technology company. I am very pro and bullish on an acquisition as a great exit for a company.

That being said, why would a company IPO? They should IPO if they believe they have a long-term independent business, that their business can scale and grow independently as a segment and as a portion of the market and they have the ability to go do that.

Second would be they want access to the capital markets, beyond just the venture capital and the funding market. They want access to the public capital markets to continue to grow and scale their business. That sounds – On the one hand, that's great. You get access to a much larger pot of cash that you can go tap into. But it also comes with a price, and that price is more regulation, more scrutiny, more internal processes that you need to put in place as a public company. So any company thinking about going public has to be aware that many of those things come along as the benefit of the bargain if you go public as well.

[00:47:06] JM: So I'm pretty naïve about an IPO process. I know that during an IPO, a lot of the people who founded the company have some shares that they get to sell and they get wealthy. Does the company itself also sell shares and they can get a bunch of cash in their coffers? Is that what you mean by access to the public liquidity markets?

[00:47:28] HC: Yes. The primary reason that company would go public is to raise capital, to raise money, just like they would raise from a venture capitalist. It's just another vehicle for raising capital. When they go public, they're effectively selling some percentage ownership in the company shares to the public market at a specific price. Then they receive that capital back into the company effectively as cash that they can then go spend as they build their business. So they are trading some equity and some shares for cash that they used to go grow the business.

[00:48:05] JM: Those different IPOs that you've been involved in, how did they differ in terms of the outcome?

[00:48:10] HC: So they were all good outcomes in terms of going public, and every one of those companies went public for the right reasons. One of them, one of the ones I was involved in very early on was back in the late 90s, it was a company called Vitro Technology, which was during we would all call the .com boom, and some of the listeners may be too young to even know what that was. But that was a time when probably valuations got well ahead of what could be delivered by the software companies. So many companies that went public ended up with lower valuations, and that being the case of Vitro is one of them, but a successful outcome overall for the company.

Most recently, the Hortonworks, I think was a great outcome. It allowed the company, yes, to get access to capital markets, but also to raise its visibility in the market to have more people knowing who Hortonworks was that allowed the company to be successful to compete against at the time, Cloudera, MapR and other companies that were merging the Hadoop space. I would say it was a good outcome. One that was worth it for the company to go through and was done at the right time.

[00:49:21] JM: So we live in a world where the cloud providers have so much impact on the software ecosystem. Let's say I'm building an enterprise software company today, and I mean maybe the early days or maybe I'm in that \$10 million trough of difficulty that you mentioned earlier, or maybe I've even made it past that. How do I factor in the existence of the cloud providers in my strategy?

[00:49:50] HC: Great question and something every smaller technology company has to think about, is how do I either partner with the cloud providers, compete against them, or figure out how to exist knowing they're out there?

In many cases, it's how do I work with the cloud and how do I best run in a cloud-based environment? A common approach and think a good strategy is, "Can you run in a public or private cloud environment with your software and can you create a managed service around delivering that software into the market?"

So a good example of that I think is a company that's doing that very well right now is Databricks. They are delivering a managed service around Spark on the public cloud and then to the market and they have come up with a good model of how do I leverage the cloud? How do I work in the cloud? How do I embrace it? But how do I make money by delivering more than just Spark? How I deliver that as a service that companies can go use? I think that is a very good model in terms of how to work with the club vendors and work in a cloud environment.

[00:50:59] JM: How is Kubernetes affecting the market for startups that are playing in this space and I guess the competitive strategy for cloud providers broadly?

[00:51:08] HC: Kubernetes is providing an opportunity for many new software companies to be developed. The reason I say that is Kubernetes as a way to orchestrate containers and go move in that space suddenly makes it easier for you to build applications into containerized environment and go scale and grow them. This is a shift just as we want for mainframe to client server, and client server PC, and PC to virtualized environments, and now it's virtualized in many cases to containers. Every time one of those major shifts happen, the whole infrastructure stack gets rewritten, and there are opportunities at every layer of the Stack to go deliver a product or a service, and Kubernetes is helping to drive and foster that innovation in the market.

Now you're seeing service mesh companies of how do you manage interactions between those containers? You're seeing new analytics companies just say, "How do I monitoring and management knowing that containers are more ephemeral and they don't last as long and they could be here and then gone?" How do I best monitor and manage that at that scale?" You're seeing a whole series of technologies being developed to go leverage that containerized

environment, and I think Kubernetes is providing the impetus for a whole new set of software companies to get developed, and that makes it a really exciting time to be in the market for those types of companies.

[00:52:35] JM: If you are in that market, the strategy that you're defining is, like you said, has to be aware of these cloud providers. You got to be – I mean, I saw that AWS announced service mesh product I think yesterday. So do you have any specific pieces of advice for these companies that are getting started in the Kubernetes space? Is there anything about this time that the Kubernetes world that leads you to giving – To think about very specific advice for this world?

[00:53:11] HC: Some things I would consider as I'm building that out is how do I get scale quickly so I have enough presence in the market. Then when one of the cloud providers inevitably offers something similar, I've built up enough capability to be able to withstand that storm. I think a company is done a good job as somebody like Hashicorp. It has a built a hybrid cloud solution that plays very well on this new container and orchestration world with a whole set of services they deliver around that. They've done a very good job of how they navigate that from.

But if I'm a new company being developed, I would want to consider, "Do I do this is commercial or open source? As open source, I could get broader distribution and usage of my product and probably better visibility quickly." Second, "Do I deliver a managed service? So I'm not just monetizing the product knowing that a cloud provider may potentially be able to compete against it or offer similar service. But if I offered a managed service, it's my expertise and my delivery of a capability bundled with the technical product underneath that I have a way to build a better mote that is more defensible as one of the other providers try to go offer that." Those are some things I would consider as a smaller company to say, "How do I defend myself as I start growing and knowing inevitably that if it is a big enough space, somebody else is going to go compete in it. That's normal competition.

[00:54:39] JM: You're on the board of several companies, and two of the companies that you're on the board of are Citus Data, which is scalable PostgreS company, and Unravel Data, which is the APM for data systems. How does your experience at Hortonworks and that time in the

Hadoop space, how does that carryover to these newer data companies and what kinds of shifts do you see coming in the changing world of the data platform?

[00:55:12] HC: So they use those. Yes, I'm in the board of Citus, a board adviser to Rnrael.

[00:55:18] JM: Oh, okay.

[00:55:18] HC: I think two obviously very different companies with Citus building a scale out PostgreS database leveraging the open source PostgreS community and the work that they do within that, and then scaling with the capabilities that companies want to build out their database to more than a couple nodes, they need a better capability. Citus fits that very well, and much of the – I think they're doing a very good job and worked with them on the board is how do you work with the cloud providers? How do you scale your business knowing inevitably they can offer PostgreS as well and have a way to go build a competitive offering that both partners win and competes in that market? I think they're doing a very good job of that.

Unravel Data I think is a great company, really focused on how do I have a monitoring and management capability for the big data stack, and they're doing a really good job of scaling the business. So with many of the companies that work with from a border and advisor level, it's providing the lessons and experience that have seen of scaling the company to the various stages and how do we help those companies not make the mistakes that are inevitable at each one of the stages? How do you know when to step on the gas and accelerate and hire more to go grow the business and when will you get a better return? And when potentially do you need to put your foot on the brake and slow down, because you don't have enough proven capability in the market and the last thing you want to do is invest a lot of your caching and run out of money. So how do you get that balance at the appropriate levels of the company so you can best build your market?

[00:56:57] JM: That Unravel Data, I think this is a pretty interesting company, because the big data stack is becoming more and more of an operational data system rather than just like this offline thing for reporting or an off-line machine learning job. It's getting more and more into the actual application logic. Then I've seen a number of different companies that are trying to attack that emergence of the operational big data world, or I guess the data platform. I mean, I've

seen these companies like Dremio, and then data breaks obviously, and it's obviously a big – Just a gigantic opportunity, but it seems like it's really, really hard. What do you see as the primary difficulties of building an enterprise data platform?

[00:57:52] HC: So I'd say a couple of things around that. So an enterprise data platform – So the first thing I'd want to think through, am I truly an enterprise data platform? Which means I am something that other things are going to go run on and get built on, or am I an enterprise data product? Neither one of is good or bad. But if I'm a product, then I'm going to want to go sell that product to the users who need it in the market.

From platform, a platform typically has a longer sale, typically a more complex sale. That is one that I think you want to think through. Do you truly want to be a platform day one, or do you want to sell a set of products that you can ultimately integrate together and become the platform for that company down the road? Those are interesting strategy questions that a company should be thinking through early on, because they may trade off the view that on this great platform that everyone can use and suddenly find that it takes them a long time to get traction in the market.

[00:58:58] JM: So last question, you are clearly enjoying your time as kind of this consigliere to a variety of different companies. You get to touch on a lot of different things, different strategies at different places and a lot of different areas of the stack. Do you ever have desire to go back to running a company or being an operator? Any desire to start a company? Is your heart set on what you're doing right now?

[00:59:28] HC: Yeah, that's a very good question. Obviously one that I constantly ask myself, Jeff, and what I would say is I really enjoy working with multiple companies that are wrestling with the same challenges that myself and Mitch and Sean and other partners, AccelG2M worked through at early stages in the company and finding it is fun to help figure out that puzzle.

I say that I've spent my career in decades in sales marketing and go-to-market distribution, but I was an engineer by education and I'd learned that figuring out that puzzle and how you get the parts together across multiple different companies is a really interesting way to continue to, I'd say, work with the market, the market that's out there today. So I see myself doing this for a

while and continuing as AccelG2M and as a consultant in doing what we are to help the company, and it's very satisfying to see where they end up as we've worked with them.

[01:00:29] JM: Herb, great to talk to you. Thanks for coming on Software Engineering Daily.

[01:00:31] HC: I appreciate it, Jeff. Thank you very much. You run a great show.

[END OF INTERVIEW]

[01:00:37] JM: This podcast is brought to you by wix.com. Build your website quickly with Wix. Wix code unites design features with advanced code capabilities, so you can build data-driven websites and professional web apps very quickly. You can store and manage unlimited data, you can create hundreds of dynamic pages, you can add repeating layouts, make custom forms, call external APIs and take full control of your sites functionality using Wix Code APIs and your own JavaScript. You don't need HTML or CSS.

With Wix codes, built-in database and IDE, you've got one click deployment that instantly updates all the content on your site and everything is SEO friendly. What about security and hosting and maintenance? Wix has you covered, so you can spend more time focusing on yourself and your clients.

If you're not a developer, it's not a problem. There's plenty that you can do without writing a lot of code, although of course if you are a developer, then you can do much more. You can explore all the resources on the Wix Code's site to learn more about web development wherever you are in your developer career. You can discover video tutorials, articles, code snippets, API references and a lively forum where you can get advanced tips from Wix Code experts.

Check it out for yourself at [wicks.com/sed](https://wix.com/sed). That's wix.com/sed. You can get 10% off your premium plan while developing a website quickly for the web. To get that 10% off the premium plan and support Software Engineering Daily, go to wix.com/sed and see what you can do with Wix Code today.

[END]