Computer Aided Design (CAD) Pioneer Workshop
Day 2 Session 6: Key Companies (1980-2000)

Moderated by:
David C. Brock
Burt Grad

Recorded May 2, 2023

CHM Reference number: 2023.0065

Table of Contents

Jon Peddie: David Brock, I'd like to propose a question. A topic that we haven't discussed that I'm curious about is electrical CAD, and I have several questions about it: A) are we going to discuss it, B) what does this group think was the company that started it, and C) what does this group think is the surviving company that's still doing it?

David Brock: Well, we talked about it some yesterday, and I think we can talk about it again today, especially when we get into the application section today, or as we're talking about key companies, if somebody wants to raise that up.

I wanted to talk a little bit about the format for the next piece of time that we have. For about the next hour and 45 minutes, we can talk about key companies, 1980 to 2000. Jon Hirschtick won't be back with us for at least another half-hour, so what I thought we could do is have Carl [Bass] kick us off with his remarks about his personal experience and his contributions, with maybe have a little discussion around that before moving on to Michael [Payne] for his individual discussion and then some conversation around that. Then Jon H. will be back with us by that time. We can turn to him and then open it up for our general discussion. But I'll write down eCAD to try and return to it, JP, either in this session or the next one.

Peddie: Okay, and we still have constructed geometry to get to also.

Brock: Yes. So, with that, Carl, if you'd like to take it away as they say. I think you may still be muted.

Founding of Autodesk

Carl Bass: Okay, let me first qualify this by saying the founding of Autodesk, which was around 1982, was before my time working at the company, but I'm obviously familiar with it. I spent a lot of time there. In my former life at Ithaca Software, I sold software to Autodesk, Computervision, PTC, etc., all down the line, so I know something about it.

The origin story of Autodesk is that it was started by a group of mainframe programmers who saw the opportunity for PCs. Somewhere between 17 and 20 people founded this company. They at some point went to a tradeshow with several products. One of them was this proof of concept by Mike Riddle that has a previous name, but eventually becomes AutoCAD. In fact, the reason the company was called Autodesk is one of the other products at the tradeshow was an automated desktop. Think of the 1980 version of Microsoft Office or Lotus Suite. That's what they were thinking about. But they went to a tradeshow, and they presented the office suite and the CAD stuff. The CAD stuff got a much better reception, so the product became AutoCAD. For 40 plus years, it had the name Autodesk, even though it doesn't you know? It's this forgotten piece of history that never materialized.

Just to set the record somewhat straight: In lots of literature, and we even refer to it today, John Walker is listed as the founder. I think that's really unfair to the 17, 18, 19 other people who all threw in some money and did a fair share of the programming of what turned it into a commercial product. It was an early version of the Silicon Valley company. It was egalitarian and libertarian, and it had a whole bunch of quirky individuals who were the founders of the company. They always joke that the CEO was the person who was the least capable programmer. I mean certainly John had a leadership role, and I'm not trying to diminish what he's done, but I just want to make sure that everyone appreciates all these other people who quit their day jobs, just like what happened at Fairchild, Intel, and a number of other stories. They went off and said, “I'm going to take hundreds of dollars to invest in this thing, take my time, and start a company.” Just to put this in perspective. For the several thousand dollars John Walker put in, he still owns, I believe, something like 5% of Autodesk. He still owns 5% of this $50 billion something company.

Joel Orr: And the reason he is in Switzerland is pure taxes.

Bass: He hated paying taxes for the company and personally. That was just part of his libertarian philosophy.

Orr: Yes.

Bass: He was originally doing hardware designs under a company in Sausalito called Marinchip, and that led to the other. The one thing I would say that's really helpful, at least for the history, is John was incredibly articulate and very prolific, and much of the history was contemporaneously recorded in something called the Autodesk Files. It's still online. He wrote these so-called information letters that he sent out periodically, and it describes in a very transparent way both the technical and business challenges or questions facing the company. Not many other companies have as explicit a history that was recorded at the time. I like it because it doesn't have some of the flourishes that come from writing the history after the fact, where all the ugliness and whatever is smoothed out, sanded, and polished. It really has all the junk in there and kind of the real-time thinking about it.

Peddie: It has the emotion. He wrote a terminal paper deciding how Autodesk was going to die.

Bass: Yes, many times. I wrote one too.

AutoCAD and Early Diogital Design

Bass: So AutoCAD, at its inception, and primarily to this day, was a 2D drafting program. It was something that allowed people to record and document the kinds of designs they were doing. It was a replacement for the drafting table, and it merely did it electronically. You could preserve it and get all the benefits from digital stuff. What was interesting about it and different than some of the other threads we've pulled over the last day and a half is it was more horizontal. People used it to do mechanical design, but they did it for electrical design, landscape design, architectural design, and civil engineering; it could be used for all kinds of disciplines because the common form of communication at the time was a piece of paper or vellum with lines, arcs, circles, and text on it, so it was broader in some sense than the other things. The other things had higher sophistication and complexity but were less broad. That was the trade-off.

One of the things that I think attracted customers to AutoCAD was that the price point for it was completely different than what had come before. It ran on a PC in the thousands of dollar range, and the software cost in the thousands of dollars range. That was completely different than owning an IBM 360 or whatever huge mainframe. We've heard a lot from Dave [Kasik] about Boeing, which was representative of the other aero, and auto companies, with that kind of usage. But this opened up computer-aided design to a whole audience of people in small- and medium-sized businesses that never would have even thought of it or considered it.

The second thing that went with Autodesk that was also different is that it ended up being just a software company. We heard about the turnkey vendors selling hardware, software, and all the rest. AutoCAD ran on commercial IBM PCs, and it spawned an entire industry of add-on hardware and software, things like tablets. We forget about these things: graphics cards, which evolved into GPUs [graphics processing units], and floating-point coprocessors when the PC was not very capable of doing floating-point math. This all evolved around it and allowed people to push the performance to some high fraction of what was available on either mainframes or what was becoming Unix workstations at the time. And it stayed there.

Another thing it did that I think has lived on with other products: It was customizable in its user interface, and it was programmable with an API with the use of a language called AutoLISP, which was a derivative of the LISP that was developed at Cal Berkeley. We talked this morning about the DWG file format. Probably just as important was the DXF file format in terms of getting information, not to other CAD systems, but forfabrication. To this day, I'm sitting in my machine shop, where if you look, at least half the machines still take as input DXF files. So 30-something years after the advent of DXF, there's beautiful, very high-end equipment that's running off this fairly simple file format.

Autodesk Competitors and Markets

Bass: We talked about the competition a little bit. There were many, many, many other capable programs at the time. We've talked about some of them. VersaCAD was an obvious comparison, but Bentley, Micro CADAM, and CADKEY came along. When you try to figure out why Autodesk won, there are certainly technical things you could point to. I'm not sure that that was the defining factor. To Jon's [Hirschtick} point this morning, what it did really well and continued, one of the things SolidWorks adopted, was to develop a way of selling the product through resellers. Most of the resellers had started as end users, and it was like some Amway kind of thing almost. I can't remember the details exactly, but the economics were to buy two copies of AutoCAD for the price of one, and the idea was you would sell one to your friend. You'd pay $3,000, you can either get one copy or two copies. Most people got two copies. They sold one to their friend. Some of those people saw this as a way to make money in the future, and they turned it into real businesses. It was a jumping off spot for businesses that sold hardware, software, training, and consulting, and I think they do to this day. Certainly as we go into the 1990s and you look at SolidWorks, one of the things it does really well in addition to moving to this new platform is it takes this business model of selling through resellers and takes it to the next level.

The other thing about it, as I mentioned, is it served many markets. Over this period, from 1982 when the product was introduced to 2000, there's very specific versions of it built for those disciplines. There's a specific version of AutoCAD made for civil engineering, and there's one for GIS [geographic information system], one for mechanical design, and one for architectural design. The entire field got more specific and catered more to individual needs because, despite the generic basis of just putting lines and symbols on paper, there were very specific things. The tolerances in mechanical design are completely different than in architectural design, so that's what the customized versions did.

As we get to the end of the year 2000, I think there's two things really of note there. The product itself was doing about $1 billion in revenue, and as we mentioned yesterday, Carol Bartz was the CEO at the time. She's probably the most high-profile executive in technology at this point, or almost the only one, and certainly, to this day, remains the only CAD CEO who is a woman. That was a fairly unusual move. Carol had her background at Sun Microsystems in digital equipment, two of the workstation makers, and moved over to do that.

Let me just wrap up with one more thing about Autodesk a number of people have brought up. Autodesk is slightly different than some of the other companies mentioned because of this broad base. From 2000 all the way to the present day, it ends up really in three end-user markets. In those, it's really the dominant player. It's an also ran in other areas such as in mechanical engineering, which has been a lot of the focus that

we've talked about today. But it is still substantial, over $1 billion business, so I shouldn't denigrate it too much. Two other areas are architecture and construction. It becomes the leader in that field, and it becomes the leader in media and entertainment, in developing software to create games and movies. To this day, both of those market leadership positions are still there.

Peddie: That was through acquisitions.

Bass: Every company is. I mean every one of these companies is. I know when I was CEO, Jon, I did over 100 acquisitions. There's nothing special about whether the egg is fertilized by you or not.

Brad Holtz: But the point overall is we haven't discussed the role of acquisitions and consolidations in the overall scheme of things.

Market Drivers and Anomalies

Bass: Let me just end with a few observations, and I think it'll lead right into that, Brad. Let me end with two economic and two technical observations. So the first one is, if you look at the history of technology, a huge amount of literature has been written by the business school gurus about how technology markets are generally winner take all or winner take most. You certainly can see that there's not a viable competitor to Photoshop. Microsoft Office dominates. There are many of these products, and mostly you end up with leaders in these categories that are untouched and far ahead of any competition.

One of the most unusual things to me, and I've noodled on this for years, has been that, while there's been a huge amount of consolidation in the CAD industry, it never ended up with one dominant player. If you look today just in CAD, leaving alone analysis, simulation, and visualization, there are four multibillion-dollar companies that are hugely successful and have been for over 20 years. Most of them go back 20, 30 plus years. It's kind of a crazy and anomalous thing relative to the other technology domains, and despite what we've said before, the longevity of these companies is kind of amazing. If you look at things like security or document management, any enterprise software thing, these companies are much faster to grow and much sooner to flame out.

Now on the product side, you do have a little bit more of winner takes all. You can look at individual areas, but even in mechanical CAD, which has been most of the focus, right now there are probably half a dozen products out there that are fully capable of doing what 99% of the users need to do, and yet, the market has not consolidated around one.

The second thing I would say is the thing that I observed over many years about this market that's different than other technology markets. Despite the fact that the software is highly technical, the adoption of the technology is relatively slow. It takes a long time to grow, and then it takes almost forever to disappear. I'll tell you a story about when I had dinner many years ago, it must have been in the late 1990s, with Mike Payne and Jon Hirschtick, and we were talking about their new little company called SolidWorks. I asked Jon about the business plan, and he said, “We're going to do $1 million the first year. We hope to double it the second. Then we hope to double it the third, and then I hope to double it the fourth, and I hope to double it the fifth.” I don't know if you remember this, Mike, but I just said, “Is that your plan?” He said, “Yeah, that would be pretty good, and you know how doubling works.” They went about and succeeded in doing at least that. It wasn't a much more sophisticated business plan than that.

I would say, about this early take up of products, the single thing we fought the hardest against in understanding product market fit at Autodesk were the false positives because revenue in the early years was so small. When a large company put out a product, it was really easy to get false signals from the market that what you were doing was interesting, when it really wasn't. But if you look, you now have a product. SolidWorks is, I don't know, getting close to 30 years old. AutoCAD, which is still close to a billion dollars in revenue business, is 40 plus years old. So even after these things have reached their top, reached the apex, the use of them degrades really, really, really slowly. So it's just a different economics. That struck me that it's one of the things that makes it different than the other technology businesses, and the fact that there's been so little consolidation amongst the major players strikes me.

Let me just end with the two kind of technical things about it. One of the things, and I made this point a little bit yesterday, is that I think what's really important is the alignment that we saw between the development of hardware and software, the underlying mathematical representations, and the eventual processes and techniques used to fabricate stuff. You look at the wonderful curved surfaces that people who made airplanes and cars did, and it winds up very nicely with what we could represent with NURBS and other kinds of splines. But it also lines up perfectly with what was on the factory floor in terms of stamping and CNC [computer numerical control] machining. So we got this really nice alignment where you could go from conception to physical artifact and preserve those shapes all the way through it.

The exact same thing, by the way, happened in architectural design. I always laugh when I drive around and say, “I can tell what building that software was designed with.” Because as soon as the product introduced curved curtain walls, all of a sudden, you drive around every city, and there's curved curtain walls on their skyscrapers. Then there's these parabolic curtains, and you're like, this is crazy that it's so influential. As the toolmakers for people doing the design, these were the capabilities we either made

easy or difficult, and as you made them easier and more approachable, more people incorporated them into their products.

Computer-Documented Design

Bass: My last point: Many of you have heard my spiel about computer-aided design, but it's worth repeating in the history, particularly with Jon H.'s remarks yesterday about how he thought we came halfway. I don't know if it's optimistic or pessimistic. I think we're kind of in the second inning of what's possible. I don't think, to this day, most of computer-aided design has had the computer aiding the design. What we mostly have is computer-documented design. What we had starting in the 1980s were all these products like VersaCAD and AutoCAD that could do adequate 2D representations, and now what we have are 3D products that do 3D documentation. But what we don't have is this integration of engineering, simulation, and fabrication. To my mind, we're still modeling fillets before we even know if the product we're designing fits the engineering bill in terms of function, price, performance, or manufacturability. It's kind of ass backwards that we ask for a degree of specificity before we even solve the engineering problem.

I'm a user of CAD software. I use it every day to this day, and I am totally amazed compared to what I did when I worked on PDPs, Evans and Sutherland's, Grinnell frame buffers, and ran Paddle 2. I would intersect a cube with a sphere, go and get coffee for 20 minutes, and come back and go, "Wow, it made a hole in my sphere." So I am totally amazed at what we can do. But I also believe it's way short of what's possible. I think there's a whole new generation, and I feel fortunate to watch young, smart people who are inventing new underlying mathematics. They're inventing new processes and materials for manufacturing stuff and whole new ways of designing things. I'm mostly optimistic about seeing what people are going to do, and I think it will take that same alignment of software, manufacturing techniques, processes, and mathematical representations to get the same kind of success many of these companies got in the 1980s and 1990s.

Digital Twins

Peddie: Carl, do you have anything to say about digital twins?

Bass: It's marketing BS. Look, people for the longest time have tried to get high-fidelity digital representations of the thing they're building. Whatever the heck they want to call that thing, I don't give a shit, truthfully. Here's how I'd say it, Jon. What I've wanted since I first started in this thing is to be able to design a product and essentially turn it on. I want to, behind the glass, hit the button and have the thing work. I want to know what it sounds like, what it looks like, how it performs. When I have that, then I

think we've fulfilled the dream of what's possible with doing digital representations of physical artifacts.

These terms like digital twin, there have been dozens along the way, and they're useful, I think, for the user community to have a conversation. But the fundamental idea of having really accurate representations of what you're going to build early in the process and being able to understand the performance, I think is really important, as I've said a bunch of times. Anyone who's ever built something will tell you, if they knew at the beginning what they knew at the end, they would do it differently. What digital technology allows us to do is make digital mistakes much more quickly and cheaply than analog mistakes.

Peddie: I was saying, with regard to digital twins, is traceability for safety.

Bass: Yes, it's certainly important. There's no doubt, but the entire list of stuff is important. There was a lot of talk about moving information around and file formats. The thing to me that's always been more important is the thing that I think Mike pointed out:

I hate the fact that the tools are non-associative, that the changes I make in one place don't go anywhere else. It was either Mike or Jon that pointed out the referencing systems are different, so I have taken a vow. I will never use a non-associative tool. Life is too short for me to make a change over here and then have to make the change over there. I only have time to make one change, and the one thing that computers should be good at is propagating those changes everywhere and tracking those changes.

Peddie: Omniverse is going to solve that problem for us.

Bass: Omniverse is full of shit too.

Data Longevity and Legacy Software

Brock: Well, I think we have Dave, and then Peter waiting to get in on this.

Dave Kasik: Carl, that was a marvelous recounting of Autodesk, and I have three comments. Number one is about the world of digital twins and the like. I'm going to send out a link to something that the Air Force Integrated Computer-Aided Manufacturing program did in 1977 with a character named Dennis Wisnosky. Anybody know Dennis?

Holtz: Yes.

Kasik: Dennis has this great picture of a guy sitting at a computer terminal and an airplane flying out the butt end. One picture. It's pretty interesting, so I'll send that link

out. The other is about the use of mechanical CAD. I think this also applies to ECAD, Jon: Mechanical CAD and other artifacts that are created during the design process for both engineering and manufacturing. Certainly, that is a big deal and the use of that data for downstream processes is essential in companies like Boeing, and I would imagine in most of many others. Getting that data in a form that can be used however is nontrivial because there are differences in representation for finite-element modeling, aerodynamics analysis, propulsion, and everything else. The manufacturing world is just the same. Just because we have the data doesn't mean it's necessarily useful or usable downstream.

Last thing is about the comment that you made, Carl, about the longevity of companies. I contend that is far more about the longevity of the data than it is about the longevity of the companies. You get somebody sucked into a highly proprietary format, and the technical term is that you're screwed because you can't convert it easily into another system without painstaking effort.

Bass: It's almost impossible.

Kasik: Exactly.

Bass: I would say the one thing I'd just add, Dave, is I think data is the key. It's kind of the muscle memory and the learning and training of the organization. The third thing is the integration with the other enterprise systems.

Kasik: Absolutely. So you build that spiderweb of dependencies, and all of a sudden, it's really hard to get rid of stuff. I'll give you an example from the Boeing Company. Boeing wrote a system in late 1970 called WIRS that controlled the configuration of all of the wire harnesses in flying products. That system lasted until about three years ago running on IBM mainframes, written in Cobol with more tricks than you can possibly imagine. I'm aware personally of about four different tries to change that, to get off that, migrate that software to something else, and they finally got it working. Not by changing the software, but by writing an emulator, a 370 emulator that will work well enough that they could keep the damn software running. So they finally got rid of mainframes, moved to a Unix environment, but it's still the same software. It's amazing how we are both blessed and cursed by our legacy.

CAD User Groups

Brock: I think we have next, Peter, followed by Michael. Then I think I saw Brad wanted to get in.

Peter Marks: That was terrific, Carl. I have one comment and one question for you. The comment is maybe more for the people trying to track the history. You've noted this

winner take all and there certainly are a lot of markets where that's true. But there's a whole bunch where it's not true. In fact, probably dominant where it's not true, and that may explain the CAD history.

I've done a lot of research on how customers make buying decisions across all kinds of industries. I did that for the CAD industry with a book that turned out to be *Aligning Technology for Best Results*, where I visited 20 companies and asked, “How did you pick your system?” The general rule for people is that we love to make binary decisions, good or evil, left or right, etc. If we think we're doing a really good job of choosing things, we test drive three cars. We pick three vendors to fix something for a house. Then we throw away one and pick the two. So there's this human psychology that kind of drives us to two or three people being active in the choices.

If you look at the CAD industry, there's this market that serves the Boeings of the world, the big companies that listen to them, and I would argue that Siemens and Dassault are the two giants left standing, and then you've got the masses, the democratization. Autodesk is king of that. It’s a little bit like Microsoft Word or something. But you've got SolidWorks, now a part of Dassault, still trying to have it both ways, and it's a tough organization, so there’s that. Then you've got PTC kind of in the middle. A lot of companies trying to have it both ways for the big companies and have the lower product.

Anyhow, I think, if you want to track what's happened to companies, what's happened to markets, how we've gone from Brad Holtz’ 200 competitors in the CAD rating guide, down to two or so in an industry, that begins to explain it. So that's the comment for the history.

The question I've got is that I had a chance to talk to a whole bunch of user groups, and I think at Autodesk University, and I’d f like to know when that started. The SolidWorks user groups were pretty unique in that these were not only inspirational events for the users, but it became a mechanism for smart users to feed ideas back into the company. Maybe it's outside the 2000 frame, but I was hoping you could talk a little bit about just your mechanisms for listening to all these guys with great ideas who took it into different fields and so on. How'd that work at Autodesk?

Michael Payne: I think AU goes back to the early 1990s.

Holtz: CAD campus was in the 1980s, which was the precursor.

Payne: I think the first AU is in 1992, 1993, or something like that.

Marks: When does it become Woodstock?

Bass: It was one of those things that we figured out. It was very different than selling to the gigantic companies. The function for the company was clearly marketing and promotion of its products. But the way it did it was subtle in that it made it a learning experience for all the users, and to this day, it's a hugely successful thing. I think Jon H. would tell you the SolidWorks thing was a copy of it, to some degree, and it's hugely successful to get people working in these individual companies together, year after year. I learned a lot about how to use the product better.

But what I really did is I networked there, and I found other people who looked like me working in another company doing something similar, and I learned from them. Before all of this learning was shared online, as it is today, you had to physically be in the same place, and it was a great way to get that. It was hugely successful. I can't think of any other marketing activity that perhaps was so valuable as that.

Payne: Just a quick point about Carl's meaning of CAD, which was the French are always different. For those of you that don't know, CATIA actually stands for a jumble of French, unpronounceable words, which mean interactive design in three dimensions assisted by a computer. I think you made that point, Carl.

Bass: I appreciate the CATIA name.

Product Lifespans

Holtz: So three points. There was a conversation earlier about Boeing's lifecycle. The planes are lasting 50 umpteen years. You need to keep those original products that they were designed with up and running so that you can access it. I do think that the lifecycle of what is being designed—and in buildings that's probably 40 years—influences much of what's happening with product transition.

The other is there's a huge difference in that most of the folks here are coming in general from the manufacturing or mechanical side, but on the AEC [Architecture, Electronics. Construction] side of the world, one of the fundamental structures that's different is that there are many parties involved that are keeping information proprietary and information about the product, which in this case is a building. They're designing different pieces of it, and nobody owns the whole thing, and the people who are funding the investment into the design are not necessarily benefiting from the results. That has a big influence on what's going on there.

One thing that I would like to bring up is, in 1984, there was a seminal meeting founded by Terry Wohlers, the MicroCAD Forum, and I think he ran that for three years. That's where I actually met Joel [Orr] for the first time, and many of the players in the industry who became significant portions of the industry, either as consultants, vendors, or whatever, participated in that. That to me was the launching point of what eventually

became a group of other communities, including CAD camp, and so on. Anyway, you can't forget about that one event.

Kasik: I'll make one really quick comment, Brad, about the lifespan of a product. The more important part about having data longevity is that you can actually use the same part on different products. So it's just as important for the longevity, keeping inventory control, and keeping reuse as a possibility to have that data longevity. It isn't just about the thing flying for 50 or 60 years. It's about making parts available across time and across different product lines.

Holtz: If you think of the plane as an assembly, the parts that are being designed, the lifecycle of the individual part, is just as important as the lifecycle of the assembly.

Kasik: Absolutely.

Holtz: I absolutely agree.

Founding of PTC

Brock: Michael, if it's okay with you, maybe we could shift our attention now and give you a chance to share some extended remarks about your experience.

Payne: I definitely can't be as exciting about it as Carl was, but well, first of all, it started when I hired Vladimir Geisberg to make a 3D modeling product. You laugh, but I think you may have run into him or something.

Kasik: Indeed.

Payne: I remember a conversation with him, and I asked, "What's your brother doing?" because I knew he had a brother. He said, "Oh, he's wasting his time messing around up in Andover or somewhere." Anyway, Vladimir was supposed to do solid modeling better than at Computervision, and it limped along, and it did something, and he hired Leonard Rice to help him with geometry.

Later, I left Prime, and sometime not long after, I get this call from somebody who could hardly speak English. In a very strong Russian accent, he said, "You don't know me, but you know my brother, and he said I should talk to you." I'm not making this up. I go and have breakfast with the guy, and he takes me to see a young guy who used to work at Applicon. Sam [Geisberg] worked at Applicon at one time, and they showed me a PCAT with a block, a bracket with a hole in it that had dimensions on it. You could move it by changing the number on any dimension. I looked at that and said, "Well, that's what your brother should have done for me at Prime, but he didn't."

The PCAT at that time was a complete birdbrain. It had no memory. It had no disk space, and it had nothing else. So that was about all you could make on a PCAT. Sam and I got together, and we found a VAX, one of these big things in a cabinet, and a Techtronix display, and we made it work on the VAX. Now you've got more memory and more disk, and it was a bit faster, but not that much faster than a PCAT. We went around the entire venture community in Boston, and this story is that PTC nearly didn't happen. There was no money. Actually, Sam had run out of all the money that he had and he forgot that you have to pay taxes. I don't know what you do in Russia, but you have to pay taxes in America.

Anyway, Sam was a math professor in St. Petersburg when he was in Russia, and we go to all these venture guys, and they learned words like NURBS. He'd say, "Do you use NURBS?" if they had an expert. Sam would say, "I could explain to you what we do, but you wouldn't understand," so we didn't get any money from that guy. We went around to everyone in Boston, including Don Fedderson from Charles River. Don had been the CEO of Applicon. He wasn't technical, but he kind of believed that Sam might know what he was up to. Eventually Fedderson and Felder Hardymon at Bessemer got together and said, "We'll give you some money." But before that, we sort of went on holiday because we didn't have any money, and this is true, and I found enough money that Sam had hidden away to pay his taxes.

After a few months, the lawyer called us and said, "I'd like you to meet this guy." It was somebody called Young. We talked to him, and he said, "I don't know what the hell you two are talking about, but I believe you do, so I'm going to give you $300,000, and I'll get some friends of mine to give you another $300,000." So now we had $600,000. We hired Leonard and rented an office where we had to take the furniture, so I told them, "You have to give me a reduction in the rent, so that when the VCs come and talk to us, if they do, they won't think we're spending lots of money because we can't have this fancy furniture. You have to take it away." We got a huge reduction in the rent so we could say that we negotiated a good deal.

Anyway, we started building a product, and it was clear that you could not do what Computervision and some others did, which was sort of just model this geometry, and then if you wanted to change it, you had to model it again. In other words, it was show and tell. You can't build products with show and tell. We knew you had to not only make it change, but we had to make it larger and make it change. So you did things, but you had to make everything. There was nothing on the computer, but it was what you could call a utility. We had to make our own solver. We had to do our own storage. We had to make our own geometry. I'll come back to geometry.

We had to do our own double byte for Japanese models, so we could sell in Japan, and we had to do our own macros and our own graphics because there was no product like

HOOPS around we could use. But I did find a textbook called Foley and van Dam, and it had the answers on how to build graphics. Didn't it, Carl?

Bass: Oh, absolutely, Mike.

Payne: My story of PTC [Parametric Technology Corporation] is at the beginning and end of a cycle. The beginning of this cycle was that workstations were starting to emerge, so Sam and I had a very heated discussion on whether we were going to build this product for workstations or for minicomputers on long pieces of wire. I won the battle, and it was workstations, so we made it work on the VAX. We got some VAX workstations, which were relatively inexpensive compared to the minicomputers, and then we found a used Sun at a going-out-of-business sale. So we made it work. Now we had a problem because it was a different byte order, so we had to make everything work this way and that way on byte order. We did, but that wasn't so easy. The answer was to store everything in ASCII or a form of ASCII, kind of like DXF.

Anyway, along comes some money, complete with Steve Wolski and Dick Harrison—they're names that some of you must know—and we found another business. Apart from selling the software, there were lots of Unix vendors of hardware, and they all had a variation of Unix, which was supposed to be standard. Some looked like BSD Unix and others didn't. We had a lovely business charging these people to port our lovely software to their horrible hardware. The first victim was NEC; that paid $300,000 to put it on their computer. I don't think there's anything confidential in what I said there.

It became obvious that nobody was going to buy this software without trying it. I mean, how do I know it's going to work? How do I know you'll be around and all the other stuff. One of the notable failures was when Dick Harrison went to Raytheon, and the engineers really loved it at Raytheon. Then Dick goes to purchasing, and the guy says, "I hear my developers love your stuff. Well, we're not going to buy it." "Well, why are you not going to buy?" "Because we only do business with a hundred-million-dollar revenue companies," and Dick said, "Well, if everybody took that attitude, there wouldn't be any new hundred-million-dollar companies," and the guy said, "Yep, you're right." Then Dick said some unmentionable words, and he never did sell anything to Raytheon until that guy retired, which was about seven years later.

So this became remarkably successful. It was that rise that you were talking about, Carl, going from double to double to double. Then it went public, and it got rather boring, to be honest. In 1994, we had marketing people, and it was like the *Hitchhiker's Guide to the Galaxy* or the restaurant at the end of the universe that was full of accountants and marketing people. They thought they knew what they were talking about.

So I go to this meeting where Microsoft came. This is in 1994. Actually, it's 1993, I think. Yes, it was the end of 19'93. They had some presentation about some new operating

system—they didn't have a name for it. It was going to be a 32-bit operating system, and I listened and didn't say anything. I didn't say anything until the next day. Then I called the guy and said, “Can you give me that operating system?” and he says, “Well, you didn't say anything yesterday.” I said, “That's because I didn't want anybody to think I might be interested, because if we’re interested, it wouldn't happen. You give me a copy of that, and I'll send it to Israel and have a guy in Israel work with it.” So we produced a version of Pro Engineer that ran on Windows NT, ready for the next Autofact. Then one of these marketing geniuses came to me just before the Autofact and said, “Do you think we could put on some demonstration for Autofact?” I said, “No, we have a product that was ready to ship.” We took a PC with Windows NT on it, and this was about the same time as they launched it. They called it, what, Windows?

Holtz: NT 3.1.

Payne: Yes, it was NT. It was actually ready to go, and the marketing people kept hiding it, so nobody saw it because it would upset the sponsors of the hardware, these big boxes from the Suns and the Silicon Graphics. That was about the time I felt I have to go and do something else because you knew they would not move to the next generation of stuff unless they were forced to. In the end, they did, but it took them a while. So that's kind of the story up to that point.

There's two ways you can do 3D geometry. One is it can be 3D centric, which is Parasolid and ASUS. The other is that you can be 2D UV centric, which is Pro Engineer and CATIA V5, V6. One can have a debate all day long as to which is the better way to do it because there are pluses and minuses of each. That may explain some interoperability issues.

ATP and its Effect on Venture Funding

Brock: Let's open it up. Peter, I see that you want to get in here.

Marks: Michael talked about the conversation where they debated on workstations versus VAX or something like that, and I want to share a footnote in history. In 1988, I was feeling a little stuck at SDRC, and there was a company called Automation Technology Products on the west coast that basically had the idea of PTC before PTC. It was parametric form. If you looked at the product specs, it was everything that PTC wanted to do probably about two years before. This company had gotten a first round of venture funding and had been kind of stuck at it. They asked me to come and help them get funding and take it to the next level. I thought, “Well, I need to make a change. We’ll do this.”

ATP had been founded by Bob Benders and John Benbow. Bob was the president of Calma, and John Benbow and most of his core, John Powdea, and stuff like that came

out of ICL, the UK competitor to IBM, and so he had this great software idea. They put a team together, and I didn't think too much about at the time. I thought, “We’ll plan our way out of this.” They wrote the first iteration of the software in PL1, and because they knew that IBM was taking over the world, destined it for the VM operating system, and I was there.

We got funding. We had a who's who of customers. I actually had development sponsoring for a pretty successful NC verification system. But I remember the board meeting in which we decided what we would do next. We had bugs to fix. There were problems and the rest of that. There were about seven of us sitting in a room and a sales guy who was probably selling to Dave Kasik. IBM was one of our 12 major customers, and the decision was, “We’ll be okay if we port this to MVS.” I'm going, “I think that maybe we want to fix the bugs before, rather than port them to MVS. The documentation for MVS is bigger than the room that we're sitting in.” Anyhow, it was six against one that we're going to port to MVS. Russ Hanke came on. He sort of missed that one too. That was a company that probably read the same tea leaves in terms of the software, but completely misjudged the direction. It was certainly true that IBM wanted us on MVS, but it wasn't the direction that the future needed to take.

Brock: Michael, did you have a reaction? Or should we let Carl jump in here?

Payne: I don't have a reaction other than, well, it didn't really happen.

Brock: Carl.

Bass: So two things. One is I'm glad they brought up ATP. It had one interesting side effect. Because at the time it was such a large venture funding, it really soured west coast venture capitalists on anything to do with engineering software. It’s like trying to do the second Theranos or something like that. For a while after the Newton came out, people said, “Oh, it has promise, but it was a spectacular failure.” ATP was the same way. I remember people trying to raise money for all kinds of things in engineering software, and that was the marker, like absolutely stay away from it.

Parametric Technology

Bass: The only other thing I just wanted to add is, and someone mentioned this yesterday, parametric technology was spectacularly different than everything that went before it. It was the thing that everybody's jaws dropped at the trade show or customer meeting in which they got to see it because it was the beginning of the fulfillment of a process where the computer would do more than merely record you mousing around and doing stuff. It was actually doing something in the background akin to an Excel spreadsheet or a Lotus spreadsheet automatically recalculating. The fact that there was a recipe there that could be recalculated was a fundamentally different way because

every prior system to some degree was mostly a mimicry of putting pen to paper, and it recorded it in various different ways. Mike may be underselling it a little bit.

There are a couple of these moments when I look back on history. I remember at SIGGRAPH when Pixar showed the Luxo lamp film, and you just know the world's going to be different. With the introduction of Pro Engineer, the world of design engineering software was going to be different from that point forward.

Payne: Let me respond to that, Carl.

Bass: Hey, I finally said something nice about you, and you're still going to argue with me.

Payne: I'm not disagreeing with you. Going back to the comments about NURBS. Just for the record, all of the analytical surfaces are kept as analytical surfaces and not in NURBS, so it isn't NURBS for everything. The same is true in Asus, and I know it is in Parasolid, and the only reason for that is the simplicity that the earlier speaker mentioned, Ken. But it's slower, and you worry about speed when doing these lovely geometric modelers. Maybe you don't care so much today, but the speed was very, very important. We didn't have the computing power that you have today when we built it. It got better and better, admittedly.

Jon Hirschtick: I just want to build on and agree with what Carl said. Even more profoundly, in my opinion, what was so significant about Pro Engineer was that for the first time it was possible to create 3D solid models of real products and assemblies in what I call a meaningful volume. I would argue that had been a defining vision of so much of the research community, so much of what CAD was being sold as, but didn't really do.

In fact, I have a brochure copy here. I could share my screen if you wanted to. It talked about it from a decade before Pro Engineer. So there's this empty vision and then Pro Engineer comes along. So yes, it's great that it was feature-based, parametric, and associated, but what was most important was that it worked. It could really be used by a real person to do something, and I really think that again, I get it. Other people had made solid models before then, for sure, but no one had wrapped it in a practical solution, and there were so many details of the implementation from how they did the underlying geometry, which we could debate and argue, and I have my own opinions, philosophies, and all that. But that's not exactly why we're here. We're here to tell the history that was so significant that Pro Engineer shaped—pun intended, I guess—to make a 3D solid model of a real product.

Payne: Yes, but that's another interesting point because way at the beginning there was another thing. Sam insisted that there was no need for drawings anymore.

That was until I ran into a company that said, “I'll buy it if you have drawings.” So we made two rotten attempts at making drawings, one called Pro Draw and the other Pro Draft, and there were a few other Pro things that floated around. But that same mistake was not made in SolidWorks. The real end of the story was after I made the thing with the guy in Israel on Windows NT. Then I left, and somehow I met Jon.

Hirschtick: That’s another story.

Brock: It's a story that I think we can turn to in just a moment. Daniel, did you have a question that you wanted to get in?

Daniel Llach: Thank you. Yes. It's a question that builds on some of the points that were just made about this. Like Carl, you mentioned that there was this moment where you kind of saw parametric modeling and realized that the world was changing. I think that's very interesting. I wanted to ask the group in general about the idea of these object-based models, this transition between thinking of CAD systems as environments to make drawings to thinking of CAD systems as environments to build the models of things. This is sometimes referred to as semantic modeling where we have more structures. I wanted to get people's thoughts about that.

Payne: That was the idea, but the world relied on drawings and still does. I give them to people to approve—the building inspector or the transmittal to a manufacturer. That's why you find all these symbols on the drawings, which people developed over the years, to give all of the information necessary from a flat piece of paper to make something in 3D. That has not gone away. Yes, some people annotate the model and all the rest of that, and maybe they do more of it. Maybe that's more acceptable. But I'd love to see somebody have a model of a building and get a building inspector in to approve it. When they talk about BIM, I think that's not going to happen for a while.

Holtz: Actually, it already has in several jurisdictions.

Payne: You said several. How many are there in the country?

Holtz: I don't know that, but I know who could get you the answer. It's not pervasive and it's only in counties that are well-funded.

CAD Sales and Marketing

Holtz: I still want to just get in one comment about PTC. As groundbreaking and game-changing as it was, one of the things that is particularly notable about early PTC was the marketing and sales activities. Their marketing people and their sales people knew how different it was and how impactful it was. What's the right way to put it? Hubris, or some other way of describing it. I don't know. Joel, can you help me here?

Orr: They were notably proud of the product.

Holtz: That's kind.

Orr: I'm trying to be kind.

Payne: A different perspective is that Dick Harrison created a model of how you go to the customer: you first of all ask for this and then you ask for that. If you were a salesman, you better follow Dick's model. Now part of that model was if a guy didn't buy, then you told his boss that he was a something or other. I think that's what you're hinting at, Brad. But since we're being recorded, I won't exactly say who the guy was. And that was somewhat effective.

Holtz: It was quite effective. PTC grew extraordinarily fast in the early days, not just because of the technology. Anyway, I just wanted to get it on record that that was impactful.

Payne: Yes. There was a point of time though at which they lost interest in CAD to some degree, when the emphasis was more on IoT [Internet of Things]. We created two offices in Israel, and they actually saved the product. That was a time at which there were a lot of smart Russians who had migrated to Israel, and we took advantage of that.

Holtz: Russia does play a very big role in the development of the technology side through all of these. It's not really something we've discussed.

Payne: Oh, yes. Actually, at least half of the people we hired in Boston were from Russia. And obviously, in Israel three-quarters of them were Russian.

Brock: Before we turn to Peter, let me interject for just a second. Jon H., while you had to step out, we had an opportunity to get extended remarks from Carl and Michael. Let's hear from Peter on this discussion, but then maybe we could turn to extended remarks from you, if that's okay.

Hirschtick: I'm happy to.

Brock: Great, thanks. Sorry, Peter, for the delay.

Marks: I had two things, one of which was a segue to Jon and kind of a footnote in the geometry history. Carl made the point that the tools influence what you can design, so if you've got Dassault and Gary uses it, the building's going to look different than if you're designing it in Wireframe or something like that. It's post-2000, but Parametrics was a revelation for a lot of people. Great for a family of parts. Probably around 2000, in terms of variational geometry stuff, a lot of people just wanted not to

have to visit the history tree, but to directly modify what they're doing. We might pick up that thread later.

The other thing that I'll say is about PTC. I was out in the consulting industry, and I had an opportunity to keynote one of the user conferences. I still remember the users talking to me and saying, "We love the software." They were blown away, but just like Jon, everybody else said, "And we hate the company." I can tell you some of the reasons. Walsky got up and his keynote was basically the same he would give to the financial analysts: "We're doing great. We're making a ton of money. We're so profitable." And the guys are going, "But what about us?" It was driven in a different way.

One of the other revelations for me was a chance to go to a SolidWorks user conference, which was Woodstock. This was part of a bunch of lessons. I'm just looking at a couple of things. One is that I learned from Chuck House], who was at the time VP heading up a technical project at HP, is that it takes three times to get anything right. So Windows 1, Windows 2, Windows 3—now it's kind of working. Same way with this thing. Jon is the software avatar of that. I first met him when he was doing his individual product, when everybody was really cool, but not quite. He did it at CV [Computervision], and then he did it at SolidWorks and got it right. He put together a great team. But the thing that was exceptional at the time was the devotion to the customers. Customers really felt they could trust this company, and he democratized it. Carl did the same thing at Autodesk; that was more constrained around there.

Anyhow, PTC got the technology right, but I think the customer interface wrong. They even had contracts that said, "You cannot use the software to diss us. You cannot use it in a demo that says anything negative about the company." That was how anal they were about a bunch of stuff. Hirschtick's company was just like, "You guys, we want you to do great stuff. Come and show it to us. We'll learn from it."

Brock: Jon, tell us about how you got there because it was a pretty marvelous environment. It grew from zero to 18 years old under your leadership.

Founding of SolidWorks

Hirschtick: I'm happy to tell you the story of SolidWorks in whatever length here. I'll go a little long because it seems like it might be interesting. I also have a presentation I can show excerpts from that will show some artifacts of the founding. Anyway, I'll tell you my story. I'll tell you how I got here because that shapes what happened.

I grew up in Chicago. I started programming computers in the 1970s. Actually, I learned both computer programming and paper-based drafting. I learned both those skills in the high school where much of the movie "The Breakfast Club" was filmed. Okay, little sidebar: I wasn't going to mention this, but since Peter talked about treating people,

my father taught me a lot about how to treat people. He had what today would be called a side hustle selling stamps to collectors. He took me along and treated customers really well, ethically, and all that. That always was wired into me, that that's what you do, and it's still to this day what I do in my work.

Anyway, I got into CAD in 1981. I was very lucky to meet Dave Gossard, and I worked at the Cab-Lab. I was manager of the Cab-Lab. I met Dave Anderson from Purdue. I want to give him credit. He is a lesser known guy than Dave Gossard, but his kind of partner. Around that time, we got a Lisa computer, and I used a Lisp machine. Got very indoctrinated in the superiority of the graphical user interface. The first Mac I'd ever seen was the one I picked up for the CAD-Lab. Gossard was all over that.

When I worked at the CAD-Lab, we had two rooms: the lab room where the terminals were and the computer room that had a raised white floor and big minicomputers. Most of you know what I'm talking about. If you don't, imagine several subzero refrigerator-sized cabinets of DEC VAX's and so forth. Gossard would bring visitors to the lab, and this is the luck I had being around him. He'd say, "This whole lab is set up to emulate what every engineer will have on their desk in the future." People just couldn't get their minds around that. They would say, "What are you talking about?" Maybe some of you visited—Joel, Peter, Mark. Gossard would say, "This whole lab simulates what will be in a computer for the desktop," and he said, "and it may not even be a special computer for engineering. It'll just be the computers people have," and people were like, "What? No, come on. I've seen those PCs. They're not worth anything, and we can't run engineering on them." But Gossard really believed. There's some videos on YouTube. I'll skip them, but if you want to see, there's a video demo of Gossard showing stuff in the 1980s at the CAD-Lab. I was in the room when that video was made. He's on camera, and it shows some really cool demos of the Dan version.

So I started a venture capital-funded company out of the CAD-Lab, and I was sold on Windows as a platform. This was in 1987. We didn't sell a lot of it, so Computervision bought the company.

I'm skipping over a chapter. Back to Computervision, where I'd been 10 years earlier. Now it's the early 1990s. I'm back at Computervision. I'd seen Pro/E demonstrated at tradeshows, and I'd seen AutoCAD kick ass in the market. The CAD market that I had known for all these rich stories that we're all recounting here was totally eclipsed. It was like everything else became archeological digging or sideshows. It was AutoCAD and Pro Engineer in those years.

At Computervision, we were getting our asses kicked, and I was tapped to be on a PTC attack team, ironically, since we weren't attacking them at all. They were taking our customers. They were all talking about going to visit our customers, and I raised my hand and said, "I want to go visit our lost customers." They all looked at me like, "What

do you mean?" I said, "I want to go visit the customers who are using Pro Engineer." A guy named Dan Monette and I got in the car and drove around. I would go watch people use them and say, "Show me what you're doing." I was like, "This stuff really isn't just at tradeshows. It's really being used."

I got kind of tired of working at Computervision. I left. By the way, my letter of resignation is published on the internet. It's part of an MBA- book, *The Portable MBA in Entrepreneurship*. I'll skip it. I decided to leave and at one point I said, "I never want to be in the CAD business again." But then I got to thinking, "Hey, Pro/E is the way to build 3D solid models, but the UI is way too complicated." I mean, the ridiculous menu cascading, and plot and don't plot and all this. It was too expensive in Unix workstations. I thought, Windows PCs are going to take over the world. I saw a future state, and I said, "When the world uses solid modeling, it's going to work like PTC. It's going to be priced like AutoCAD, and it's going to have a UI like Microsoft Word or Excel." You know, duh!

A lot of other people had the same freaking idea, so we started working on a prototype. This is in the fall of 1993. If you like, I'll show some slides here to guide you through the events that happened next here. This presentation I just dug up, and I really haven't looked at it in a long time. It's from 2005, the 10-year anniversary party for SolidWorks. Mike Payne may have been there? Were you, Mike? [editor's note: Jon Hirschtick shows a number of slides which he refers to in many of the next paragraphs

Payne: I was there, yes.

Hirschtick: Maybe some others here, I don't know. In the fall of 1993, this is the cofounding team. There is one person not shown here who had left, so there were six of us: Mike, me, Bob Zuffante, Scott Harris, Tommy Li, Constantine Dokos. Tommy had been my thesis advisor. When I had the concept, I started in my house writing a prototype. I got ACIS and D-Cubed and a Windows kit. A friend of mine in Microsoft sent me the Windows development kit, which was a big box of shit—you know, this huge thing arrived. I started working on a prototype.

Oh, this is my old house. This is where we founded Winchester Design Systems. This slide is about who was there at the event itself. But this is where I started, in that room over the garage. That's where I was writing the prototype. We really wanted every engineer to have it. These slides are 20 years old. We wanted to build a strong business, and Peter and I said at the time, "We're going to show people that we can build an awesome business while treating our customers, resellers, partners, employees, with dignity and respect and honesty." That was super important. I hope we more or less did that well. There's a lot of people who think I was an asshole too, so I don't know.

In 1994, this is stuff from the prototype. So we started working on a prototype. I had these sketches. Fortunately, I captured this years ago. This guy, Bob Zuffante, wrote coding conventions. Scott Harris did all this testing, which concluded that Parasolid was way better than ACIS at modeling. This is a little sketch of the feature manager, which we patented. That came directly out of visiting Pro Engineer customers and watching them use the thing. They had a notebook. One guy had a notebook. I said, "Show me a model." He said, "Okay," and he gets out his notebook and he flips it open. I'm like, "What's that?" He goes, "Oh, this is my feature list. Every time I make a feature, I add it in or take it out." That led to a patent on the feature manager.

We had a prototype. But we couldn't raise money. Functionality outline: I think there's functions in there that SolidWorks still doesn't do.

Getting the company going. Here you go: "PTC released Mike Payne from the noncompete agreement. [Oops!] August 1994." Our team is working along. I left Computervision in August 1993, so we were working along for nine months or so. We have this prototype, and Mike gets released from his noncompete.

The venture capitalist, Axel Bichara, I give a lot of credit to. I'll just tell you the story of venture funding. Carl was right. Nobody in the VC industry wanted to hear our story at all. They thought it was a terrible idea, especially competing against PTC. It was viewed as a crazy idea. I got turned down by like 15 venture capitalists. One guy believed: Axel Bichara. He believed. He introduced me to Mike. We raised our money. There was a business plan that we had written, and we had a sales goal of $3 million. We did way beyond that. These were other names we had looked at for the company.

Here's stuff from the original presentation. First solid model. It's a Windows product. Key selling points: Cost-effective, easy to use, Windows. That's still true today, if you ask people why they bought it. This is a term sheet for a venture capital round. Vic Leventhal joins SolidWorks, may he rest in peace. How many of you know Vic? I know that many of you do. Vic was an icon in the community, and I feel we can't mention the history of CAD or SolidWorks without mentioning Vic Leventhal, who tragically passed away. This is our founding team working away. These are photos, you don't need to see these. Sorry, I didn't edit the presentation at all. Oh, there's a guy you know, Mike Payne.

Steve Krug is a famous author of a book called *Don't Make Me Think* on web design. Very interesting guy. That's me, a younger version of me. Dave Corcoran, who was so critical to building the product. These are early screenshots, by the way. People said it was ridiculous: "You'll never get a model that will work on Windows." Now they all are on Windows. These are people who were on the team who were still there 10 years later. We got an order. I'm going quick. Okay, here you go: Computer Aided Design

Report. Talk about newsletters. I flew out to visit Steve Wolf in his house, and I flew out to visit Dave Weisberg in his house, and Mark Halper. Brad, did I come to your house?

Holtz: Yes.

Hirschtick: Joel, I came to your house, right? I wasn't just sending copies or anything. I got on a freaking plane and dragged the freaking workstation. I went to Steve's house, and Steve said, "Will you leave me a copy of the software to play with?" and I said yes, which wasn't exactly what it was used. He wrote this article, "Move Over Pro/E: The Next Generation Has Arrived." Now today you'd see something like this and say, "What does it mean?" In the moment, in 1995, that meant a lot. Then we launched the product publicly, and at the SolidWorks reunion, I said the same thing. Yeah, we were 10 years old and we could have been all full of ourselves. We instead tried to say, "Look, our history doesn't mean anything if we don't keep following through. Most of the work is yet ahead. Let's put 3D on every engineer's desktop, and, you know, most of the work is yet ahead." I was saying the same thing after 10 years at SolidWorks. That's a little bit of SolidWorks history.

Obviously, I skipped over some of the details, but I think that gives you an idea of what happened. Sales were just better than we could have imagined. One thing I glossed over is we embraced the reseller channel. But the key innovation was to take what I said: the formula was Pro/E's modeling, AutoCAD's business model, pricing, the way of selling, more or less. We could get into the differences and Microsoft's platform. It turned out timing works too. I had believed 32-bit Windows would be coming. They were working on it, and NT was out there, but NT was not what we ultimately really wanted. We wanted a real 32-bit core Windows product, and we knew the graphics cards and the graphics libraries were coming. We crossed our fingers, and it all kind of happened.

So that takes us up to the closeout, our designated chapter of CAD history here. The year was 2000 by that point, and SolidWorks is rolling, doing well. It was not the intentional result, but the fortunes of PTC had begun a decline a bit. That’s something that I joke about all the time, which I'm not doing here. But the point is that it made a mark on the market, and the best measure of the market is to see all the products people have made with it. I can go on beyond 2000. There are more chapters we can get to, but I'll close it out there, and with that, try to keep it focused. I'm happy to take questions or to hear your comments.

Licensing Deals

Payne: Well, one thing was modeling. Jon correctly pointed out that it started out with ACIS. It couldn't do the rounds. It couldn't do this; it couldn't do that. Spatial would say they would fix it, but there was a copy of Parasolid, and you saw that comparison.

So what were we going to do about it? Because Parasolid still wasn't good enough, okay? So I called Dick Harrison, and that business plan that Jon showed, we told him all about it, and we said, "You've got a pretty good modeler there, Dick. We'd like to license it." Dick said, "Well, we've never done that," and I told Dick there was a first time for everything. He said, "Yeah, but we might be competitors." "We just showed you the business plan. We're not planning to compete with you, okay?" And he said, " but if you compete with us, you know what we're doing." He thought I didn't know what he was doing. He said, "We'll demand a benchmark." I looked at Dick and said, "What will you say when you lose?" and he changed the subject.

Hirschtick: I want to be super clear. That was one of many things we investigated that we didn't use.

Payne: Yes, exactly.

Hirschtick: It's a neat story, but we used Parasolid.

Payne: Yep.

Hirschtick: We looked at other kernels too. I could get into a lot of things. There was also a lot of business deals to be done. I spent a year of my life working with Chuck Grindstaff. I would have to mention him. May he rest in peace. What a great guy. Working on a freaking licensing deal, because the licensing deals that they handed me for Parasolid said $3,000 a copy, and I'm like, we've got to sell ours for $4,000. Not going to work. Not to mention his whole design. So I could tell you, you know, there's millions of stories. We did look at a lot of kernels. We looked at a lot of alternatives but settled on Parasolid and D-Cubed. Scott Harris could show you the book. He still has a red book of his modeling test cases—if you know Scott, very thorough—and that would show you the answers.

Windows Computing Environment

Brock: Brad, I know you have limited time, if you want to get in.

Holtz: I have a conflict for the rest of the afternoon, but I did want to raise one point. One of the key changes in the industry was the transition to Windows, but the transition enabled companies like SolidWorks and Autodesk to stop having to write all their own drivers for everything, from graphics cards to plotters to printers; that load was being picked up by the operating system. That was an enabling shift in transition that allowed the companies to focus on their core technologies. With that, I'm going to need to cut out. It's been a pleasure and delightful and really wonderful seeing all of you again.

Marks: Good to see you, Brad.

Marks: Just a quick footnote to Brad's thing. Probably post-2000, companies like Autodesk also benefited from the Windows ecosystem in terms of database software.

Peddie: Just to augment what Brad just said. The companies didn't get off for free with the establishment of the operating system picking up driver support. The companies like Jon's and Carl's and so forth, they still had to certify those drivers to make sure that they worked and didn't break because if it broke, the service call came to Autodesk, not to the card maker or to whoever might have written the driver. There was still a fair amount of overhead they had to support to keep things working.

Hirschtick: Well said. Both sides are true.

Payne: Jon mentioned Vic. Vic was really instrumental in establishing the VAR network, and in that, he had help from being known to many of the Autodesk resellers and had further help from something called R13, which some of you may know about. I think Joel knows about it.

Ken Versprille: Yes, I still use the R13 story too when I talk to startup companies, saying, "You get one chance and the quality better be good." Because a particular R13 killed the idea of we'll fix it later.

In Person versus Online User Communities

Marks: I've got a question. It's jumping ahead a little bit. We had, both for Autodesk and for SolidWorks, these amazing customer events. We had online training and we had dealers. Now that world is completely gone, or almost completely gone. Nobody goes to conferences anymore. Nobody, in parentheses. I'm wondering if anybody sees that as worse, better? I mean, I think we've lost something and gained something, and maybe we pick it up later on. I want to know, what supplants that to keep the energy as well as the learning and coming down the learning curve and stuff like that going?

Peddie: What we're doing right now. This is what replaces it.

Hirschtick: Well, that's part of it. But Peter, there's Van der Spek, Pacific Events. The SolidWorks event, while maybe not as strong in attendance nor as focused in tone and content as it used to be in the past, is still a pretty strong event with thousands of people going. I believe the Autodesk events are huge with thousands of attendees.

Last time I went, I met Carl at one of the events. Carl, I don't know if you remember that. I don't think I was going to be featured on stage, but it was interesting to walk in the

door and look around. In two weeks, we're having PTC's LiveWorks event here in Boston. Even post-pandemic, we're going to get about 5,000 people—I don't know the exact number. By the way, if anyone's interested, I can get a free pass for any of you guys who want to come. I'd love to have you here. Come on over to LiveWorks. Those vendor events are very much alive, Peter.

The events come and go in the world relative to user needs. What's happened that you have today, that you didn't have years ago, is you have much more vibrant online things. You can go to YouTube and find all kinds of videos for any of these systems today. You can go and find forums that are extremely active, with incredible amounts of online activity that you didn't have in the old days. So times change, and the vehicles change, but there's still a lot of people going to shows. They're just different shows.

Peddie: You just ask Joel why he started COFES.

Orr: We wanted something that wasn't a display of hardware and as much of an opportunity for buying and selling. Of course, we can denigrate that, but we felt that there was an untapped need for a place where people could come and do more of what we have been doing here, at this thing that CHM [Computer History Museum] has called a webinar. We were reluctant even to have a keynote speaker. I have to thank people like, of course, Peter and Jon H and Jon P, who were early supporters of our efforts. It did what we wanted it to do. We didn't make money off of it, but I have to say, that was our own fault.

Peddie: But the premise that Joel said, which I liked and resonated with in the early, early days was, we want to have our conference of hallway discussions. The idea was that you're walking down the hallway, and you see your friend that you haven't seen for six months from the last conference. "Hey, by the way, whatever happened to...?" and ba-dum, something magic happens.

Orr: Yes.

Post-2000 SolidWorks and Cloud Computing

Brock: We have a break coming up in just a few minutes, but I think it would be nice to break past the year 2000. Carl kind of talked about bringing us closer to the present in his discussion. Jon H., would you like to maybe roll the story forward past 2000 and get us more to the present?

Hirschtick: I can, but people just heard from me and they may have had enough.

Marks: No, you're on a roll.

Hirschtick: If you like, I can. So, fast forward into the 2000–2010 timeframe. SolidWorks and other systems that look like it really start to bring solid modeling from thousands of users to millions of users. That is a rough way of describing the impact. That's really, really exciting. Now I can get into my personal story of what I've been working on, and I can tell you about OnShape, what I think it represents, and why we did it.

So I was at SolidWorks, and we were acquired in 1997 by Dassault Systems. I stayed 14 more years, and I continued to visit customers. Common theme: just like in the days with Computervision and the PTC attack team and the Gossard's lab and all those things, I would see the problems they were having because the good news was that they ran on Windows computer. It did solid modeling and run on Windows computers. The bad news is, I'd go to customers, and I'd have two hours' worth of getting the hardware right. Jon Peddie, great point. You know, "Well, we got the recommended computer with the recommended driver, but we happen to have the wrong service pack of Windows and the combination didn't work." You know, all this. That was one part.

The next part was installing, downloading, copying software, license codes, service packs, and updates. Who's on what version? Upgrades of software. Nightmare, okay? Adding people to teams. We didn't envision SolidWorks being used by big global teams that are distributed. The teams, as you know, are manufacturing. They're not all one organization. Said, "We'll get everyone a seat." Well, who's everyone? You know, your vendor, customer, contractor. Every day, that group's changing.

Then there was the data. The database—David, again to your point about databases—we can argue about certain exceptions, but the database we wrote to was C:\. It was files. It's a joke, you know. The database literally was C:\. Now a lot of systems would say, "Oh, well, we're not a file-based system. We use a database," but they're writing to files and they're just putting files in the database, copying files. In order to work on the model, everyone needed to have an installed, complex piece of software and a complete set of the files copied to their world. You had copies everywhere. We tried to fix it. I tried really hard to fix these problems. We built PDM system after PDM system. We built PDM systems that we never shipped, and some of them were shown publicly, so I don't mind talking about them. We tried very hard, but no matter how you do it, it comes down to software and files copied all over the place, and a dog's breakfast of equipment. Customers were really in pain. I wanted to go visit a customer and spend hours hearing about the cool products they're making.

At the same time, I was tinkering around with other things and I saw what was happening with the cloud, web, and mobile, things like Salesforce, Workdays, NetSuite, and Google Docs. Basically, everything cool. There was this growing generation of people who assumed that the whole world would and could work that way. I also came to realize that the world of files produced an information flow that resembled paper.

While CAD had automated a lot of things, basically the information flow with SolidWorks, and I would argue that all systems like it really emulated paper information flow. We even used the nomenclature of paper:“I'll email you a copy of the file, or I'll put that file copy in a folder.” This is the terminology of paper. The icon for your email program is an envelope. I don't care which program you use. They all use an envelope. That's a paper artifact for a reason. It's a copy and you move copies around in a shell game way.

Whereas, when you look at business systems, like all the ones I mentioned, they don't copy data. Salesforce, if you went into the accounting department and said, "Hey, everyone in the accounting department, here's how you work. If you need to put a charge in the general ledger, lock the general ledger, and then download a copy of it to your local workstation. While you're downloading the copy, nobody else can make any changes, okay? And then we'll put the copy…" If you wanted accounting departments to work that way, they'd laugh at you. They'd say that's ridiculous, but that's how almost every system in the world works for engineering.

So I said, "We can solve the problems people have with this SolidWorks class systems, but we have to use cloud, web, and mobile technology to the max. No compromise. It's got to look like, work like, Salesforce. It's got to be a Salesforce-type architecture, for lack of a better term." The data lives in one place. It'll never get copied. There'll be no software to install. There's no chance of anyone being on different versions because there's only one master instance of the system. We believed that we could solve the problems people were having. Everywhere I went, people wanted to be more agile, they wanted to work quicker, they wanted to save money, they wanted to be more innovative. All this stuff about the files works against that.

By the way, several other people on this call, some of the people who built CAD systems, and I have probably had a million people come up to -- us and say, "Thank you for building SolidWorks. Thank you for building CAD." I'm sure this has happened to Mike. I'm sure it's happened to Carl. I'm sure it's happened to Ken. I'm probably missing someone here. I've never had anyone come up to me and say, "Thank you for building that PDM system." Or do you want to know what no one's ever said? "Oh, we got our design done with so much more innovative features and so much faster because we locked files, because we checked them out, or because we installed a PDM server." You're all laughing, but come on. This is the reality. PDM kind of sucks.

Founding of OnShape

Hirschtick: Anyway, we said, "Hey, we're going to build again." We founded OnShape at the end of 2012, and we started from scratch in a conference room with some of the same people, Tommy Li, John McElhaney, who was a big part of building SolidWorks. We didn't mention him. Carl was very supportive and enthusiastic:"Go build your

system." I hope you don't mind me saying that, Carl. And other people in the room. Joel, I seem to remember talking to you about it.

Anyway, it was kind of crazy. People told me the same thing they told me about Windows: "No one's ever going to be interested. What about speed, and what about security?" and all that. We built a brand-new system, a total cloud-native architecture, essentially built like Salesforce.

Today, we think that architecture is going to be significant to the history of this industry too. Is it? It is for the millions of people who use OnShape. They seem to think it's important, but we're still not the majority part of the market. We haven't made the impact that some of these other systems have made, but we've got a pretty loyal group of users from large companies to students and teachers, and we think it's a product that needed to be built.

By the way, it runs on mobile. It's a revolutionary idea that there are a few people in the world now who think that people want to work on mobile devices, like phones and tablets. I know it's kind of a niche idea—I'm joking, of course. Everyone wants to work on mobile. We're the last group on Earth talking about this: Should something be based on the cloud, and will they use it on phones and tablets?

Anyway, OnShape was acquired by PTC, just to complete some of the history, in 2019. I'm at PTC. I'm in the PTC building right now. Am I the only one in the room here who is, as of today, still working on this shit? Building CAD software? Going to meetings? Because I love it. It's not shit to me, I love it. So, I am still in the game. I am still going to meetings about the user interface for new features and which features to build.

Bass: I'm much further out than you.

Generative Design

Hirschtick: You're a retiree, right?

Bass: No. I wish I was. I'm still a daily CAD user and more doing it on generative design. I spend every day working on generative design stuff, which to my belief, is the next wave.

Hirschtick: Yes, we're all in favor of generative design, man.

Peddie: Yes, but you can only build it with additive manufacturing.

Bass: Not true at all.

Hirschtick: No, not true at all, Jon. We're doing generative design for machine parts.

Bass: Part of the generative design constraints are the manufacturability event. I have all kinds of algorithms where you say, this can be machined in a 3-axis machine or a 5-axis machine.

Hirschtick: Yes, absolutely. All the above. Generative design and AI are super important.

Bass: Jon H, you missed my little speech, but my thing was that I still think most of what's being done is documentation, whether 2D or 3D, and we still don't really understand the functionality, the performance, and the manufacturability of the parts we make without moving it around to 17 different products. You said we might be halfway there. I'm about half of where you are. The people who succeed us in this industry will get it to do.

Brock: Hold on a sec. I know Michael wants to comment. Then Burt had a question, and Peter wants to talk. Then let's take a little break after that. Michael.

Payne: Carl is right. We may only be halfway there, but there is a flipside. What I think we're seeing is the software does so much, whatever it is, whether it's analysis software or design software. People stuff things into computers and the answer comes out of the computer and they believe it. I mean, why are you learning how to program when you're a mechanical engineer? Why do you need to learn SolidWorks or OnShape? Because any fool can use it, can't they? Even people like you and me can use it. But I think we're not educating people to think it through and know how it should be before the computer refines it. Garbage in, garbage out. I think we're encouraging that even more than we used to, and I think that's a danger.

Grad: I had a very short question. Carl, what is generative design?

Bass: Just as we've seen in the generative language models, when you're looking at ChatGPT, you ask a question, you get an answer. Generative design actually goes back several decades. Some of the earliest work was done at NASA and Boeing. With what I call documentation, the idea is in my head, and I'm going to somehow put it into the computer.

Instead, imagine if you asked the computer, "What I need to do is build a "fill in the blank." It needs to have four holes here. It needs to be made of this material and fill this space envelope, and here are the loading conditions it's going to be subjected to." You can do this in structural, optical, or electrical domains. Any of the domains that it's possible to do it in, certainly fluid dynamics. For example, I need a heat exchanger that fits this space, and here are the loading requirements on it.

It goes off and literally runs tens of thousands, hundreds of thousands, maybe even millions of iterations in an optimization loop to get a nearly optimal answer to the question. It's not taking the answer that's already in your head and putting it in the computer. It's taking the question that's in your head, the problem you're trying to solve, and putting that in the computer. For the first time, it’s using the computer to actually solve the problem as opposed to record your answer.

Grad: Thank you.

Peddie: That's a great answer, Carl. Congratulations.

Kasik: I'll add just a little bit more. There's an entire scheme of design optimization that is the analog essentially for generative design. Basically, it takes what you start with and actually makes it work through all of the various analysis and engineering and manufacturing pieces.

Marks: I had two quick things that I wanted to add. One was mostly answered about the question about generative design. The RAF went way back to analysis, generative design, and things like that. Back in the group technology day, they were programming for generative design and the rest of that. I'm not sure what we're seeing here.

Brock: Jon, did you mean to be sharing your screen. Jon H.?

Hirschtick: I thought I'd roll a video. I'm just showing you one of the coolest products I've seen that has literally hundreds of generative design parts in it. Not a single part that's paraded through the town square of the research department, but a real machine that's being manufactured in quantities of thousands that looks like an industrial machine you'd recognize. I think these applications are very profound to see happening. This is on YouTube. You can look at it here.

Carl did a great job explaining it. I just thought I'd show some images of what these parts look like. They almost look organic in their design.

Peddie: They do look organic, and that's one of the high points of generative design. They approach organic shapes, which are disturbing for some people.

Hirschtick: From a history standpoint, given that history runs up to the present day and moment, we have in recent history people really building products and we're seeing a lot more of it happening. This is a fantastic, cool new thing. We can talk about the future too, but this meeting isn't about the future that I know. It's about the history, and generative design is indeed part of the history because it's out there today being used, and it's very promising for the future.

Grad: Do these need 3D machines to build them?

Hirschtick: No, they don't need it. To Jon's question: You can use generative design to design parts that you mill on a milling machine or whatever process you have.

Brock: Peter, did you want to finish your thought?

Marks: Yes, I had two things. If you want to trace the history of generative design, you've got this analysis track of optimization, and that goes back to the 1960s. You've got GT, group technology, with kind of like code, and obviously the parametric sort of fits there. You've got to explore lots of alternative approaches, which we were just talking about, and looking ahead, you've got this AI thing. I'd point out one caution: Just like the movie industry wants to do, "Rocky," "Rocky 1," or "Fast and Furious 1," "Fast and Furious 2," even the AI stuff is basically tweaking what we've already got. There's some pros and cons in that.

Software Licensing and Pricing

Marks: The question that I wanted to get to is the cloud security thing I think we've gone over. This is really for Jon H. and Carl. Even though I'm way out of the CAD industry, I still talk to a bunch of users, small machine shops and stuff like that who go, "I hate the move to cloud licensing." The reason they hate that is that they've been through 2008, a year in which they could barely afford to keep the doors open. They want to own the software and have it there in the lean times. They don't mind upgrading when times are good and stuff like that. Those people don't trust Autodesk not to jack up the prices when they've completed their total world domination. They also don't trust PTC probably. I'm wondering, Carl and Jon, if you have ideas of how you treat the customers well enough that they can trust you, and to get the advantages of updating the software. Cloud has all kinds of advantages, but it has this, I don't trust these guys, kind of a sub rosa thing.

Hirschtick: Yes, it has that concern for some people, and you're right. Those people should go and buy the large number of still available, perpetually licensed software. If they feel that that's a better choice for them and their business, that's why there's a free market. They can go buy that. Great. They're not my customers. You want something that runs locally or in offline mode, or has file copies on your local computer? Go buy that. You can go buy it from someone else, not from me.

Peter, there's choice in the market, and people should choose what they think is best. I can just tell you that there's way more customers who are happy to be free of licenses and so forth. By the way, people who think they don't need to update software should understand there's this new thing called computer security.

And it may not be a good idea to run versions of software that are five years old. God bless. They can go choose one of the other products. They're not my customers. We're not going to compromise what we're doing to cater to every little view in the market. Just like with SolidWorks, people would come and say, "Oh, it needs to be on Unix, and we'll buy it." We'll say, "You should go buy something else."

Payne: Yes, Jon, I hate to violently agree with you, but if you look at the history of the CAD companies, and you could start with Computervision for example, that's exactly what they did. It was a fixed price and annual maintenance and the rest of it. One day, the sales will level off a bit. You can't survive in that environment. You go downhill rapidly. It will be like having the phone company: you buy the line from your house and forever, 100 years, you can use it. They can't survive that way. They put in the wire. That was their capital expenditure. Now you get to rent it every month. I really am violently agreeing with you, Jon.

Bass: Yes, and Jon has to be a little bit more diplomatic than I do in my semi-retired state.

Payne: When were you diplomatic, Carl?

Bass: Really never, but...

Hirschtick: I've never tried to be diplomatic.

Bass: No, no. But the truth is, that train has kind of left the station. Regardless, there's two aspects of it. There's whether the thing runs on premises, on the cloud, or some hybrid thing, and then there's the way you pay for it. There is barely a piece of software right now that you don't pay for with a subscription. That train has left the station. Yes, Jon is absolutely right. There are instances of every kind of software you can buy, stuff that runs on the desktop and doesn't have a subscription and is a perpetual license, but it's a historic artifact, I would say.

Peddie: It's also perpetually broken.

Bass: Yes.

Payne: Yes, that's right.

Bass: Yes, look: Being in the positions we were in, we all get letters from the crappy customers, but I think you have to just look at the mainstream direction and some of the directions we outline, which is the software is moving to the cloud. There's more group collaboration that's going to be paid for with subscription, and over time,

there will be more machine learning, generative AI, and all of these systems. That is absolutely true.

Peddie: Yes, and if you want a perpetual license, do not buy a Tesla!

Brock: Let's do ourselves a kindness and take a short break. Let's maybe come back in 15 minutes.

Log Files versus Transaction Management

Peddie: I'll go first. Jon H, you were talking about the benefit of your system and the fact that there's no one making and passing copies around. I wanted to clarify that there is a copy. It's called the log file, and that's where you keep your revision data. If you modify the file, if you modify the drawing of the model or whatever it is, and then I come in and modified it and I screwed it up, you need to undo what I did. You can only do that because you have a log file, which is a copy.

Hirschtick: I'll just tell you, Jon, that no, that's like saying that every time you enter a debit, if you use an accounting system, and you put a charge in QuickBooks or something, or better yet, NetSuite, you are copying the entire general ledger of a 7,000 person company.

Peddie: No, not the entire. No, you are copying the log file.

Hirschtick: You're not copying revisions. You're recording the transaction in the database. I don't call that a copy.

Grad: That is a transaction.

Hirschtick: In other words, you say, on this day, on Tuesday May 2, at 3:27, Joel Orr entered into the list, into the account of charges for software, a $1,000 subscription to one of the civill packages. That charge is a transaction. It's not a copy of any sort. It's just a transaction. My point is this is a difference that makes no difference. The point is the user isn't shipping around copies of 1,000 files to each desktop. That's what I'm talking about.

Kasik: That part does work. It's implicit product data management. We decided early on that we were not going to talk about product data management, but in order to make computer-aided design work for multiple users, you need to have something more than sneaker net or email net. That starts getting into the world of product data management, which if implemented properly, is not visible to the user community. That's the real trick.

Hirschtick: Yes. I've never seen one. The only one I've seen is not invisible. Ours is not visible to the user community. Our users will tell you. If you go online in G2 Crowd and look at reviews, they'll say that OnShape data management is easier to use than no data management. And in fact, everyone who's ever used OnShape has used data management.

Kasik: It hasn't happened yet, so I came to that realization when managing software. If you try to manage software as a set of files, it's a debacle. It took a while to show the value to the group that was trying to do large CAD software development, that it was essential to have configuration management because you can do all sorts of stuff that you couldn't do otherwise. I think it's great for OnShape to do.

Peddie: Remember, you still have to be able to undo the damage I did to the model.

Kasik: That is inherent to any good data management system.

Hirschtick: Yes. Anyone who uses a transaction based system expects that.

Kasik: Exactly.

Hirschtick: Anyone who uses a CAD system can only succeed with that. In any of the other CAD systems I know of, that's only achieved by keeping copies of files around.

Peddie: You can see the effect of this. If you look in a Word file, for example. Let's say it's Word file that you're using on Google Docs. You look at the Word file and a bunch of people start to touch it. The actual word count is not changing, but the file size is getting larger.

Peddie: That's due to the revision table.

Kasik: Yes, absolutely.

Hirschtick: Of course, it's called adding data. Just like if I add a second part to the system, the data size generally goes up if I record a transaction.

Peddie: Sure, sure, but the end user doesn't see it.

Kasik: No, the end user doesn't see it.

Hirschtick: The end user doesn't have to think about it. They see one model. Just like this call is being recorded. It's not being recorded by each of us saying, “Let me lock the recording, and copy it to my desktop. I'll say something, and I'll put it back.” We're all

jumping in, and it's all going into a common data soup into the recording. We can go back to a previous state if we want, and we don't have to add. There's no possibility that any of us are going to look at a different recording, because it's not copied anywhere. You don't have to lock or copy.

Subscription Pricing

Marks: I have two questions for you, Jon. Actually, one is a thing. I absolutely agree with all three of you that the cloud is the future for all the reasons of security and collaboration and the rest. But I would urge you not to ignore that this is a pain point for customers.

Hirschtick: Oh, yes.

Marks: If you look at every industry, every industry today, the model is to put a straw into your credit card. People are beginning to feel like peasants working on a software farm. There's the newspaper, their cable TV, their CAD systems got a straw into them and what typically happens, all of a sudden, that straw instead of sucking out $100 bucks a month wants $500 bucks or $600 bucks a month. Nobody has solved that problem, that fear that customers had. Eventually, there is going to be a revolution.

Hirschtick: Peter, I agree it's a problem. It's a feeling. But when you said "people," a friendly amendment that I would make to you is that you introduce the word "some." Some people feel that way. Agreed. But when you say people as if they all feel that way, they don't. One way to look at it is as a data person, a scientist.

Look at the net promoter scores of the different products. For instance, if you look, and I don't know that you can do that, but you could do a survey. We do. We survey the market anonymously and look at NPF of different users of different products. Everyone know what NPF is? Those surveys that say how satisfied you are and how likely you are to recommend to a friend. That's the only question involved. There's a lot you can learn from that.

Anyway, I'll put an RNPF of our user community, and I believe it to be higher than any that I know of that involves installed software based on measurements that are done independently. That tells you something. There are a lot of people who feel the way you feel. I'd just say they have different problems.

Bass: Jon, you're conflating the installed software with a payment method.

Hirschtick: I'm sorry. You're right. I'll correct that.

Bass: With higher than any that I know that have upfront license fees and no recurring subscription. Right.

Hirschtick: Sorry. Stand corrected.

Bass: As somebody who took a lot of the arrows for this and was told that it would never work, I watched Autodesk transition close to 5 million customers from one business model to the other. I watched Adobe do it, and I watched Microsoft do it. It reminds me of that sign when you get a Hertz rental car: "Do not back up. Severe tire damage." We're past this point. Yes. Will there be an opportunity sometime in the future for a different business model?

Hirschtick: You could see.

Bass: If you're right, Peter, someone will come along and say, "I'm going to fix the sins of all these people who had the straw on me." I don't think it's going to happen in our lifetime. I think that train left the station, and that's where it's going.

Peddie: I want to make a very important point. Hang on. I'll make a very important point, and that is that Jon Hirschtick practices what he preaches. You will notice and remember that he ate his lunch in real time, whilst we have to go do it.

Hirschtick: It was my only choice, Jon. It was that or don't eat. I'm still working for a living, guys. I still have a full-time job

Peddie: So do I, damn it.

Marks: Go for it. Jon H. and Carl, you guys were in the democratization business. In other words, your customers tend to be the people who want the lower cost. I don't think you solve this problem by reversing the car and going over the little spikes or something like that. I think you do it by earning customer trust, and I think that is something that SolidWorks has earned. I think that's something you learned, Carl, when you were at Autodesk. You're not there, but a lot of software companies were becoming the Martin Shkreli of software, so you don't want to go there. That's all I'm going to say.

Bass: Yes.

Cloud Computing and Data Security

Grad: Is there a difference between what the large companies like Boeing and so forth would want to do as far as use of a cloud, or I guess they'll need the software for themselves?

Marks: What's the second part there? Between the Boeings and who else?

Grad: The thousands of littler companies have to use the cloud, but wouldn't big companies say, "I'm going to have my own cloud. I'm not going to use somebody else's"?

Marks: Well, in a sense they do. I mean, they'll go to Amazon or somebody like that. They do both.

Hirschtick: Companies do that.

Grad: There's sort of a mixture there—Who actually owns the cloud, if you're big enough?

Kasik: There are instances of clouds currently that are far more secure than the ones that were in place when I retired in 2016. They are government approved and secure, that companies like Boeing are actually allowing to be used at some level. The issue is inevitably, how do you manage a product, the configuration and the data, if it's in a federated or distributed set of data sources? Are you federating data across, let's say, Azure and Amazon Web Services?

Hirschtick: First, our customers don't really think about the web server. We tell people we use AWS just the way we tell them we use Parasol or something. Most of our users don't think about that. It's like asking about Salesforce and Azure or AWS. They don't talk about it. It's not in the vocab. People aren't thinking about hosting a workload, they're just thinking about getting a service. Just like now with Zoom. I don't know where Zoom is hosted. Maybe someone does. I don't. I have no idea and I don't think about it.

Now for people who have high needs of security validation like you're talking about, Dave, by all means, they need a solution that's validated. PGC is in that game. That's public knowledge.

Kasik: Right.

Hirschtick: POM for the U.S. Navy, for example. You know, not OnShape, but other products here. There's a group here that is very into that. In terms of federating, in the cloud, in the world of cloud native solutions, it's generally much easier to combine multiple systems because you do it through generally say REST APIs or something and hook things together. It becomes an easier proposition than when we're building the old style: install this, install that, install the other thing and try to get them to work together kind of stuff. But in general, we're not talking about which cloud and where in great detail. That's not really a big topic.

What is a big topic is access and data residence, and we do provide tools. We have an edition called our Enterprise Edition that has a lot of tools that aren't in our lower-level editions that allow you to do things like data residency and access so you can say, "I want a guarantee my data lives in the EU, and I want it to be accessed only within Ireland and the Netherlands, for example. You need to provide an audit trail with a high degree of certainty." That's in fact what has happened. We do that, but we don't do it by saying, "Choose your Azure." But we can read data out of the Azure Cloud or any other cloud; if you give us the REST API, we don't care about those.

Subscription Pricing Revisited

Grad: Another question. Do you make all of your money from the renewals, from the annual maintenance contracts or the processing services in that regard?

Hirschtick: Is that addressed to me or to Carl?

Grad: Well, I'll start with you, but you and the others, yes.

Hirschtick: Subscriptions, we call them subscriptions. Again, “annual maintenance contract” isn't really a phrase that is around in our world because that implies there was something you paid that was other than an annual maintenance contract. The word is subscription. I know Peter's going to say there's a segment of the world who say, "Oh, I hate subscriptions." But most of the world statistically today likes subscriptions. That's why they're buying products from PTC, which is by the way not all subscription because PTC is very big and we have some very large customers and institutions and things. But let's say, we have a predominantly subscription business here at PTC. You know, almost $2 billion a year. SolidWorks is just getting into this really, but Autodesk less so. The pure subscription business is where it's at, where it's called a subscription fee, and for most users, they don't have any problem; that's where they expect to buy it.

You'll find that, if you talk to people coming out of college, they don't know what files and folders are. They don't know what an annual maintenance contract is. They don't know what licensed code is. They don't even understand this. If companies want to use these concepts, they're going to need to train people. I go to our customers and like, "Well, it's a big training thing you teach people to use a system like OnShape." I said, "In five years, you're going to have to train people to use C:\ and file copy and Windows." There are articles on this. There are some great articles online about this. The younger people in the world walk in with their iPad, so this is disappearing.

Grad: I want this on the record because this is a major change in software pricing, which was always done with upfront and then maintenance.

Hirschtick: Yes.

Grad: The other things I think are all seat pricing, not contract pricing. That's copy pricing. Is that correct?

Hirschtick: It's all subscription; generally named user subscription is the way it goes.

Grad: Per seat?

Hirschtick: It's attached to your name. But it's not a seat anymore because most people use it on many devices. Just like the way you use your Apple subscription; it's not like you've installed it on one thing. The thing is fungible, the device is fungible, the subscription lives on, the data lives on.

Payne: It's a person.

Grad: But my point is, Boeing, you have like 20,000 or 50,000 users. Aren't you paying per person that uses it or machine that uses it? Or is it one copy, one price?

Kasik: The dominant model is an annual subscription and maintenance for the CAD software and/or a shared set of licenses. Because not everybody's using it at once. It's not a one license per person model that's been used.

Grad: But you have a limited number of licenses, don't you?

Kasik: Yes, absolutely, but there's also an annual subscription. Boeing pays a lot of money for those systems and to Autodesk because there are a lot of people who wind up using the software.

Grad: Well, that's what I was trying to remember. Back before 2000, the basic software pricing was still you paid for a license, and the question was how many copies you could use at how many desks.

Kasik: Right.

Grad: Some machines could probably use it.

Kasik: What happened is that there were license servers put in place, and there were a certain number of concurrent users. There were a certain number of people who could use those licenses.

Grad: Concurrent, right.

Kasik: And when you ran out, you were SOL [shit out of luck]. That's the technical term.

Hirschtick: Now that's a historical term that hasn't changed: SOL

Kasik: There are some things that are still viable after all these years.

Brock: Let’s end this session.

END OF DAY 2 SESSION 6

