

# Oral History of Rick Rashid

Interviewed by **Becky Monk** for the Microsoft Alumni Network

October 3, 2024

## Preface

The following oral history is the result of a recorded interview with Rick Rashid as conducted by Becky Monk on October 3, 2024, at Microsoft Studios in Redmond, Washington. This interview is part of the Microsoft Alumni Network's Microsoft Alumni Voices initiative. The goal of this project is to record the institutional history of Microsoft through the recollections of its former employees, so that the information may inform and inspire future generations.

Readers are asked to bear in mind that they are reading a transcript of the spoken word captured through video rather than written prose. The content reflects the recollections of the interviewee. The following transcript was edited by the Microsoft Alumni Network, which holds the copyright to this work.

## Interview

Rick Rashid: I'm Richard Rashid, but most people referred to me as Rick. I technically started on September 1st, 1991, and then I stayed until a little bit after my 65th birthday. So end of October in what, 2016 or something like that. I came in, I was recruited originally to basically be the first director just to found Microsoft Research. Nathan Myhrvold was the person who hired me. And this goes back to something that Nathan had originally proposed to the board of directors, which was the notion that Microsoft, even though it was a relatively small company back in 1990, 1991, that it should invest in a long-term basic research group. That was an important thing for the company to do for its long-term future and for its success. And he was able to convince the board to do that.

He looked around for various people, he got lots of different opinions. The original memo that he wrote, which was the one that went to the board that talked about Microsoft research being created, had this notion that there would be this older guy that would be wise and have a lot of gravitas, and then there would be this younger guy that would actually be running the place. And even though now I am the older guy, but back then I was 39 years old and he wound up talking to various people, decided that I wouldn't be a good candidate. And so I wound up being both the old guy and the young guy simultaneously. He didn't have to go and find two different people. But I think the notion was that we would do long-term fundamental, basic research, but that we do it in the context of Microsoft in the context of an up and coming very dynamic software company.

And that was really a very new and rather controversial concept at the time. And in fact, one of the biggest challenges I had in the early days of Microsoft Research was hiring because a lot of the people that we wanted to hire were academics. They were researchers well-known in their field, and Microsoft seemed like not necessarily a good bet for doing long-term basic research. It was a relatively small company back then. We had a couple thousand employees. We just crossed over a billion dollars in sales. Almost nobody in the academic world really knew anything about Microsoft or Windows. They were primarily using UNIX or some other operating system. And so I had to convince people that we were serious, that we were going to be in business for a long time, that we really wanted to do fundamental basic research and not just be some advanced product thing. And that was really hard. I mean, the joke I often tell about this is one of the very first people I called was one of my best friends from Stanford University. We were students there. He had become a fairly senior respected researcher at IBM, and I called him to see if he would come to join me in starting Microsoft Research. And he said, no, I mean my best friend, he just didn't think it was going to work.

Not a year or so later, he did in fact realize it was going to work, and he came and joined and became an important member of Microsoft Research. But it was hard. There was another good friend of mine at MIT, David Gifford, and this is one of these things where, again, I tried to convince David to come and David sat in my office and he said, look, I understand what you're trying to do, but software companies just don't survive. And so I just don't have any faith that you're going to be in business long enough for this to be a really viable venture. We actually made a bet it was a 25 cent bet. I am very limited in the kind of gambling that I do. It was a 25 cent bet whether Microsoft or Digital Equipment Corporation would still be in business in 10 years. This was back in 1992. Digital Equipment Corporation didn't even last in 1996. So for a long time at my office, I had this check from Dave for 25 cents with the memo of Rick was right. And I would always remind new people I was trying to recruit of that just because I wanted 'em to understand, no Microsoft's going to be here for the long term, and we're very serious about fundamental basic research, but it was tough in those early days, just convincing people it was a serious venture was hard.

Becky Monk: Yeah. Well, I want to back up because I know I've read that it was hard to convince you to come, too. So let's go way back to the very beginning. Where were you born and where did you grow up?

Rick Rashid: I was born in Iowa, a little town called Fort Madison in the southeast corner of the state. It has the distinction of being a U.S. fort. It was one of the forts that the Indians actually took back then. So it was not very successful for it. And it was a small town. I grew up there. I lived there until I was 18. And the funny thing is that growing up, the thing I really wanted to be primarily was an astronaut. I was very much interested in science and I loved building things. I just loved building electronic things, taking things apart, putting things back together again. And so that was kind of the environment I grew up in, but very small town. My parents didn't go to college, so I was really the first person in my immediate family to go to college.

In fact, my parents really didn't know very much about what I did. I went to Stanford, and that was kind of a unusual thing for somebody from a small town in Iowa. But my guidance counselor at the school decided that I would be a great candidate for Stanford, and he actually sent in my application, which didn't even tell me. I'm not sure if that's really legitimate or not, but in those days, things like that happened. So I went to Stanford. I was in the math honors program there. I was going to go on to graduate school in mathematics, but I also took all the computer science courses. There were keeping in mind that computer science wasn't a degree program back then, so it was very early days, but I took a lot of computer science courses and a professor there, drum Feldman, who was the associate director of the AI lab, was decided that he was going to start a new computer science department at the University of Rochester in Rochester, New York. And so he knew about me from my activities there. He contacted me and convinced me that, no, I didn't really want to do mathematics. I wanted to go to grad school in computer science. My parents had no idea what that was. And in fact, years later when my dad was dying of cancer, he started telling me all these stories about things that he never told me when he was otherwise doing well.

And he said, look, when you told us you were going to go do graduate school in computer science, we thought this was the dumbest idea you'd ever had in your life. This didn't seem like a sensible thing to be doing. They'd never heard of a computer. They'd never seen a computer. And he said, my mom evidently cried all night. I mean, they were super supportive on the phone call. They were great parents, but they had really serious doubts about whether this was going to work or not.

Becky Monk:

Well, this was in the seventies. Yes?

Rick Rashid: Yeah, early. I graduated Stanford in 1974, and so I went to Rochester in 74. In fact, this year they held their 50th anniversary celebration at the computer science department.

Becky Monk: So I grew up exactly straight across the state from you in southwest Iowa. If you get on your highway and go west, I'm at the very end and in a town about the same size. So were there computers in your school at that time?

Rick Rashid: No, of course not.

Becky Monk: Yeah,

Rick Rashid: I mean, they would've been millions of dollars back then. I mean, the government could afford to have a computer, but even companies, you really couldn't really have computers back then for doing anything. That wasn't an incredibly valuable activity. And the early computers worked poorly. They weren't particularly fast, they didn't do things particularly well. So no, I mean, I didn't even know about computers until I went to

Becky Monk: Stanford. So what was it that sparked your imagination about what you could do with computers and how you would actually go on to have a master's degree in computer science?

Rick Rashid: Right. Well, and PhD, basically the thing that really captured my imagination was a friend of mine who actually is the person I mentioned earlier, Dan Ling, who eventually became the director of the Redmond Lab for me, one of my best friends at Stanford, and he had taken a computer science class and he said this was a tremendous amount of fun that I should do that. And so I took the first class, which was basically simple programming. It was all done on punch cards. You'd write your program and punch cards, you'd take them to the computer center, they would take your punch cards, you would come back the next day and they would have run

your program. It might've worked, it might've not. It was a really primitive environment. It was interesting. It was a fun thing to learn.

And then I went on to take the next class, which was a computer architecture class, and there Stanford had access to some very early HP minicomputers or what we would now call a minicomputer. Very early, very primitive stuff. I mean, the way you booted the machine up, you had to key in instructions on the console. So it was very primitive, but I took this class and I remember writing a program that would read from a paper tape, sort a bunch of sentences, and then print them out on an old printer. And the first time I got that working, I think it was close to midnight, it was the most exciting thing to me because my intellect had animated this otherwise inanimate piece of hardware, something I thought of caused this machine to do something, and it was just, I don't know why. It was just really, really exciting to me.

I can remember going back to my dorm room and I think I was floating. I mean, it was so much fun. And so from that, I went on to basically take all the computer science I could at Stanford. There wasn't a lot back then, but I took what was there and then that got me interested in it, and when Jerry Feldman came to me and said, I'm going to start this new department and would you be interested and so forth, I decided, Hey, mathematics, I can do that. I could always change my mind and go back and do math. I'd been accepted at Berkeley in mathematics, very prestigious program and so forth. I said, no, let me do this computer science thing. It might be fun. And so I went to Rochester because of a relationship that Jerry had with people at Xerox Park and also the relationship that the University of Rochester had with Xerox because they were in the same city and they had a lot of the same people on each other's boards and so forth. We were the first place outside of Xerox to have the Xerox Alto. And so even though it was this very primitive startup computer science program, we had some of the best equipment in the world. And so it gave us a leg up in terms of, for

me, especially just giving us opportunities to both learn but also do things at the cutting edge that other people couldn't do. They didn't have access to that kind of hardware.

Becky Monk: I imagine that just set your imagination running of what could happen with this technology.

Rick Rashid: Well, basically these were the first full screen bitmap displays. The machine itself wasn't particularly fast, but you could program at the microcode level, and so you could make it much faster than it otherwise might be. And so we could do all sorts of crazy things. We built text editors compilers. Again, these were early days of computer science, so there were so many things that we could experiment with. We were also one of the universities that was able to get access to the Arnet (experimental local area network set up at Xerox Parc), but again, we were relatively small department didn't have a lot of money, and so I wound up having to actually design and build the boards for interfacing the Arnet to our computer network, our ethernet computer network. So it was fun. We built games. I built one of the very first network computer games called Alto Trek. It became sufficiently famous for that. Then when I was hired at Microsoft in the New York Times article about me, it referenced the fact that I was one of the people who built Alto Track. So I thought that was particularly funny, the things that you get known for.

But still, you can actually go to the Computer History Museum and you can still run that game. They have an emulators for the Xerox Alto there. And that program, that game is still there. It's still able to be run, and it had an influence really beyond what you might ordinarily think, because the fact that it was one of the very first ever computer games, I had to come up with techniques for doing things like multicast in the context of the game that then had an influence on what eventually became the internet standard, the ethernet standard for multicast.

Becky Monk: When you were designing that game, did you think it would have that far reaching of an impact?

Rick Rashid: No.

No. Honestly, so many of the things I've done at the time, you don't really think that much about what the long term might be from graduate school. I went on to become a professor at Carnegie Mellon University in Pittsburgh, and I created something called the Mock Operating System, MACH, and this was an operating system kernel concept where you could build other kinds of systems on top of that. And that became extremely influential in a variety of different settings. So it became the core kernel for what became Mac OS. So if you use a Mac product, an Apple product, an iPhone, or Mac OS computer, or if you're using an Apple Watch or even the Apple Vision stuff, you're actually using code that I wrote back in the 1980s. In fact, one of my students, Avi Tian, became head of software development at Apple when Steve Jobs went from NeXT computer to take Apple back over again. And so he brought the NeXT operating system, which really had mock at its core to Apple, and then that became the core of all these other products. But then we also, it became important to Microsoft. The concepts that we had in MACH got adopted within the context of Windows NT in the sense of this notion of having a multi personality operating system. That became one of the things that Windows NT was advertised as.

And of course, that also influenced on versions of Unix. So pretty much Windows, Unix, Mac, you're probably using something that either was influenced by or that I personally wrote. And if you asked me back in the early 1980s, if I realized, did I realize that what I was doing at that point was someday going to be running on a cell phone? I would've said, what's a cell phone? You're influencing all these things in the long future. They didn't even exist back then. Well, the concept of cell phone existed in somebody's mind, but it didn't become a real product until way later. Yeah, it is interesting to

think about your work that you do early on. I think because I was working in the very early days of computer science and developing early operating systems and games and networking technologies that it had probably an outsized influence because it just got baked into the bottom of everything.

Becky Monk: So when Nathan and George Bell started thinking about Microsoft research, the idea of having its own research lab, why do you think you were at the top of the list?

Rick Rashid: I don't necessarily think I was at the top of the list. I actually don't know what the list was, the list of people who Nathan and Gordon Bell had in their minds to run Microsoft Research. I don't know what that list was. I do know that my first connection to all of this came, I was cooking dinner at home at, this is in Pittsburgh. I was a professor of computer science at Carnegie Mellon at the time, and I get this call from Gordon Bell. So normally the phone rings it's dinner time, you assume it's a telemarketer, but I pick up the phone and it's Gordon Bell. I had known Gordon for years, his work at Digitalit was connected in with Carnegie Mellon. There was a lot of interconnection between Carnegie Mellon and Digital Equipment Corporation at the time. So I'd known Gordon really well, and I said, okay, well, I have to take this call.

I wasn't particularly convinced at this idea of Microsoft creating a computer science research organization. I knew how hard that was to do, and I wasn't sure that a small company like Microsoft would be capable of doing it or even really understood what it meant. But I listened to Gordon. He convinced me to have him and Nathan come to my office and talk to me more about it. So they did do that. And again, they didn't quite convince me at the time, but they convinced me enough that they come out to visit us, come out to see Microsoft. So I went to Microsoft and I met with Nathan. Of course, I met with many of the Microsoft executives at the time, and I was incredibly impressed with the people I met. One of the things

I think in those days at Microsoft, everybody was just amazingly smart. And I don't just mean they knew how to write software or they were technically smart. They actually understood computer science extremely well. Many of them had read the literature quite well.

They were knowledgeable both about the technology development side of computer science, but they knew a lot about research. I mean, Jim Chen, who was there at the time, I mean, I had a fantastic conversation with him and it was very technical, very focused in research, but I would still, I wasn't particularly sure then to talk to Bill. And we spoke for probably an hour and a half, I think maybe longer than was originally intended. And what amazed me about that conversation was that I sort of went in thinking, okay, Bill doesn't really probably understand what we're really talking about doing for a computer science research organization. But really, no, it pretty much every part of the conversation I had with him, he and I were very much in sync on what this meant, what would need to be done, how we need to be built, how it would be organized. I was just blown away.

Rick Rashid:

All that being said, I went back to Pittsburgh and I still told Nathan "no." And the reason was because I had a lot of loyalty to Carnegie Mellon and I built up a large research group there doing the MACH operating system and all the things that went around with that. And I just felt sort of an obligation to the people that had brought me into the department and supported me. And then it became even more complicated because they were in the process of turning computer science department into a school of computer science, and they wanted me to be the dean of the school. And so then I wound up with two different new opportunities. Should I be dean of this new computer science department or should I start this computer science research organization at Microsoft? And during

this time, Nathan just kept coming back and coming back. My understanding was internally, Nathan would refer to me as “Dr. No,” because I kept telling him, “no,” but he's really, if you've met Nathan and you may have, he's not one to really take no for an answer. So he would keep trying. And it got to a point where there was a particular day that I said, look, I'm just going to make a decision. And it sort of wound up being, I realized that part of the reason I wanted to stay at Carnegie Belling had kind of a guilt component to it.

I would feel guilty if I left because other people were depending on me. Alen Newell, who, who's a Turing Award winner, one of the key people in the department there who at the time was dying of cancer and he had been a mentor of mine, and I felt he really wanted me to stay. He felt like I was the next generation that was going to help guide the department. But at the same time, he gave me tremendous advice for what to do if I did move on. If I did go to Microsoft, I think it could be a lot of fun. So there's guilt here and there's fun here. What should I do for people who know me? I'm way more of a fun person than I am a guilt person. So I said, look, I'll do the Microsoft thing if it doesn't work out fine. But the honest truth was I had enough belief in myself that I felt that if I went there, I could make it work. And so I just decided on a particular day, that's the day I told Nathan I was joining Microsoft, and that's the date by which we begin the history of Microsoft research.

Becky Monk: Okay. So you came to that crossroads, you said no to being dean of the new school of computer science at Carnegie Mellon, and you now have to create a brand new research lab inside a software company. Help me put it into perspective. How unusual was it for software companies to have their own research labs?

Rick Rashid: Well, back in 1991, when we started Microsoft Research, really there was no other software companies tended to be fairly small and to the extent that you had computer science research labs, they were

primarily associated with hardware companies or companies like AT&T Bell Labs where they were established companies with large incomes that were very stable and they believed they could make those kinds of investments. The idea that a small startup kind of software company would be willing to make that kind of investment was really kind of unheard of. And even more to the point, the approach that I chose to take and that we took in Microsoft Research was one of doing fundamental basic research. So by definition, it was defined to not be something that necessarily applied to Microsoft. In fact, I developed really sort of a mission statement about Microsoft research. That was the mission statement we had for the entire time that I ran the organization, and I believe they still have it today.

And the first part of that was our goal was to do fundamental basic research and push forward the frontiers of knowledge in the areas that we do research. And there was nothing about that that said anything about Microsoft or even any practical use. The goal was we really needed to push this state of the art and move things forward. And my belief was that unless we were doing that, we would not ultimately be that valuable to Microsoft. In a sense, the notion here was we needed to create things that were going to be a real long-term value that could help sustain the organization in periods when it needed new ideas and new technologies and people who were capable of pushing the company in a new direction. The second part of our mission statement was once we created, developed an idea or created this technology to work as hard as we could to move that technology into use into product, but that was really the second step. It didn't impact what we worked on. It's like if something made sense, then we would put all of our energies into trying to make sure that it really did have an impact. And the last part was really to ensure that Microsoft and really our society really had a future. If you think about the role of basic research, it is kind of like an insurance policy for the future.

If the technologies change, if the economy changes, if there's a new disease or a famine, all sorts of things can happen, right? New competitors. And from a company perspective, you want to make sure that you can change, that you can adapt, that you can survive. And you do that both with the treasure chest of ideas and technologies, but also with the treasure chest of people who are the top people in their respective areas and can help lead you into new fields, whether that might be search back in the old days, search technology, whether it's technologies for software development or program analysis or for program efficiency, whatever happens, AI, whatever comes next, you want to make sure that you can change and you can survive. And this was an idea that came, you can find it in a paper written by Vannevar Bush who helped to create the National Science Foundation after World War II.

And he had this notion that technology helped to win that war. Engineers, scientists, those people helped to win that war. We need to make sure that we continue to invest in basic research. And that's where the National Science Foundation came from. And really that's the value to a company like Microsoft. And if you look at our software competitors in those days, they're almost all gone or they're irrelevant, whatever that means. They didn't make those kinds of investments, they didn't adapt, they didn't change. And I think the thing about Microsoft that I think has been probably its greatest asset over the years is this willingness to keep looking around and saying, what's changing in the world? How do we adapt? How do we lead in new areas? How do we adapt to change when we have to? And a lot of organizations, a lot of companies, a lot of people are not willing to do that,

Becky Monk:

I think, especially as they get larger, right? Especially as they get larger or older in their focus. In the case of people, what were some of the key things that you wanted to make sure as you were building out this brand new research lab, what were the things that you wanted to make sure were incorporated? I know that you

wanted to, in the middle of campus, you wanted to be not set far away from campus, but what were some of the things that you thought really would lead to success?

Rick Rashid:

Well, there were several aspects. Part of it was recruiting. We needed to make sure that we were getting the best possible people and the people we were recruiting against were universities. And in some of the other labs that existed, Bell Labs, IBM research, and anyone from those days can tell you. I mean, I was constantly pushing to say, what are the best practices these other people have in their recruiting? How do we adopt or incorporate those ideas? Or what do we do that's going to allow us to bring in people that they can't get? And I think there was this constant notion of what is the status of our recruiting? How are we doing? Who are really the best people out there? How do we talk to them? How do we bring them into this environment? What do we need to create for them in Microsoft to make them successful when they get here?

So there's the recruiting component, making sure you get incredibly good people. And I think of this, especially in the early days, we also, we focus a lot on trying to get really strong senior people that had had a series of successes, a series of accomplishments, not just one big accomplishment because we knew those people were going to keep progressing and that they could also bring other researchers in around them that wanted to be there with them. So that was a significant component to it. Another aspect was making sure that we had the right kind of environment for working with the product organization. And so one of the first things we did was to create a team, a program management team that would actually work with researchers, but also work with parts of the product organizations to try to bring the ideas of one to the other and make sure the researchers understood what was going on in the product organization, what issues they were confronting, what problems

they were dealing with, what opportunities there might be for new ideas.

But that the product organization has also had a conduit for the ideas from the researchers. And I think that was kind an important component. Again, of those early days of trying to say, how do we make sure that we do have that impact? How do we train our researchers to the product organization, what the product cycle is? We didn't make sense to work with the product team. And when it frankly doesn't, I mean, if they're just in the last stages of shipping something, they don't want to talk to you. You shouldn't be talking to them, right? They, they're busy. When they're done, then they're very open because they're looking at what the next version is going to be, what the next product is going to be. So there's lots of ways that we would try to bring that information knowledge back and forth between the two organizations.

I think that was important. One of the things I did was to establish this notion that if a researcher wanted to go work within a product team, that they would not be punished for that. Meaning they always had a golden ticket to come back, and that was always available to them. And that was important because in many cases, the best way for ideas to bridge a gap is with people, is for the people who understand them and know them to work with them. When we first started, a simple example, when we first started this idea that we would develop our own search engine at Microsoft, I sat down with the guy who was starting up the product team there, and he basically said, look, I want a list of your guys who know how to do this or your people, and these are mine, and we're going to put them in their own space working together.

And it wasn't just about search, it was also about building the server infrastructure and the technologies that would let you run things at scale. So lots of people think, oh, it's just about building a search engine. No, it's not. It's about building an entire infrastructure that

can take billions of requests and get back to you right away. So there's a lot to it. And so we had people that were in our systems organization, database people, people that were in inferencing technologies and search technologies, all working with the product team. And again, everybody knew that this was all hands on deck to get something to happen. But when it was done, the researchers, if they wanted to stay there, great, if they wanted to come back to research, they always could. And I think this notion that we would work as hard as the product teams to help realize their goals was something kind of ingrained from the very early days of Microsoft.

So one of the things that I got involved in the very early days was compression technologies for dos. So you wouldn't think research organization working with the DOS team, but there was a situation that arose when Microsoft was sued by a company called Stack, and they were claiming that we were infringing on some data compression patents that they had, strictly speaking, there were data compression patents they bought from someone else, but hey, don't want to go into the weeds too much. Anyway, this was the big problem because we were about to come out with DOS 6, and one of the key features of DOS 6 was compression technology, disc compression technology and RAM compression technology. And so this potentially was going to disrupt our product launch, which was hugely impactful from a financial perspective.

So the product teams and the lawyers came to me one evening, I think it was like 6:30 at night or something like that, and they said, we've got this problem and can you help us figure out some new compression technologies that no one's ever even heard of before, so that this lawsuit just goes away in some fashion, or at least we were immune from it. So a young researcher and I literally spent months working with the DOS team, and pretty much every day it seemed like we would come up with a new compression algorithm.

The product team would, the lawyers would review it, the product team would implement it, they would test it that night. The lawyers typically the next day said, oh, I'm not so sure this will be. And so we would just keep doing it. I think for months on end, we would just invent new compression technologies and many of which eventually became patents that Microsoft had, but it was kind of crazy. But I think people began to realize that these researchers, they would actually just, they would work as hard as anybody in the product team to make sure that the right thing happened. And we were sitting there, we developing new ideas, we were bringing in consultants to try to evaluate these new compression ideas. And it eventually led to the compression that was in DOS 6 was one of the ones that we developed. The lawsuit went on, they eventually won in the jury in the sense that a group of jurors who didn't have any education in the field whatsoever decided that this small company in California, the suit was in California, was being infringed upon by this company and bigger company in Washington state.

But by that time, we'd actually developed yet another algorithm. And so we were able to change the algorithm, continue to ship, and eventually we wound up with doing a settlement with them and it all wound up. But I think there's an example of saying we're willing to, if we have to, we'll invent on demand.

Becky Monk: I mean, it feels like that really is a great example of you guys jumping in to invent, to solve a problem that is happening right now. How much of that was done versus working on ideas that were yet to even be thought of in the trenches, I guess, of the developers? Well,

Rick Rashid: But I think there was a lot of both going on all the time. I mean, an example of inventing something that the product teams didn't initially even want. Again, back from the very early days of Microsoft during the 1992-93 period, some ideas that actually originally started with Nathan Mhyrvold and Gideon Yuval, within Microsoft

Research, they had this idea that you could compress, you could reduce the working set size of 32-bit programs by breaking the programs into piece into tiny building blocks and then rearranging them based on how the programs executed in time. And the idea was that you'd measure how programs behaved. You would run these algorithms to try to figure out what would be the optimal way of laying out these basic blocks in the programs. And then ideally, that meant that parts of the code that didn't usually get executed wouldn't be in the working set of the programs.

And so you use less space. So a number of people worked on that. I got involved in helping to drive the development of it with several people, and we got to a point where we were able to demonstrate pretty convincingly that we could have a significant impact on the working size of 32-bit programs. But I remember going to the product teams with these ideas and they would kind of look at us and say, yeah, we'd have all this math. I mean, there's just a huge amount of math on the boards. And they'd look at us and say, you guys are just really smart. I mean, you're just incredibly smart, but we don't actually have this problem because back in those days, Windows was built on 16-bit code, but you guys are really smart, and I'm sure if you keep working, you'll someday do something that we actually need and thanks for stopping by and playing our game.

That's how it mostly went with the Windows and the Office teams, except for the Windows NT team where, and those guys are super incredibly smart guys. They understood what we were doing and they had their debug code, so that wasn't the issue. But they didn't really like this idea that we were going to rearrange their software. And I remember after what I thought was a great meeting, I mean everybody, they completely understood the amount. Everything was great. We get to the end of the meeting, I thought, these guys are definitely going to want to use this. And Dave Cutler came to me afterwards and said, you're going to do this to my operating system over my dead body. And if you've met Dave, or if you met him back

in those days when he was a younger man, he was being totally serious about the dead body thing.

He was way bigger than I was. He was much more intimidating. Now, fast forward a few years, and now we're 1995ish, 94, 95-ish because of various trade issues with Asia and things like that. At the time, the amount of RAM that was available in PCs in those days had not really grown the way you would normally expect to have grown. And so we were getting into a situation where we now had a 32-bit version of Windows, Windows 95, but the size of memory on PCs of those days was sort of eight, four to eight megabytes, megabytes, not gigabytes. Keep in context for the time, and it just wasn't going to work. You weren't going to be able to run Office and still have that be a viable experience for someone. But the good news was that we have this technology and we continue to develop it. And Bill Gates and Nathan, incredibly supportive of all this. They've been supportive of us continuing to develop it, continue to work on it. We literally were doing parallel builds. We had refined the technology so that it could be used in production. And so we worked with them and we were able to bring that Windows 95 in Office 95 to market and run them on machines that were much less capable because we had this technology that reduced the working set size for by about a factor of two. I mean, years later,

Rick Rashid:

Various people would tell me, save Lotus, right? Lotus, Word Perfect. They didn't have this technology. They couldn't run in these smaller systems. It was a tremendous disadvantage to them. So it was really a technical advantage for us. And they said, well, we knew you were doing something that made your office work better, but we didn't know what it was. And I think that was something where originally the product teams didn't want the technology. They didn't think it was relevant to them, but times changed, something happened that suddenly meant that they had a problem and this was a solution for that problem. And because we'd worked on it

and believed in it and continued to invest in it, it was available to them when they needed it. Eventually for Windows NT, I eventually got a great note from Dave Cutler, which basically said, I want to thank you guys for all the hard work you put into this.

We couldn't have shipped without you. So again, that's a very simple example of investments in research, which originally was just basic research, but then eventually had this incredibly practical effect because of a change in the environment, a change in the way the world was functioning. And I think that's really where research is valuable. If you are only investing in things that you know are going to pay off, then you're going to be missing the most important opportunities. And I think that's why we would hire people in new research areas where Microsoft didn't even have products. We became one of the top 3D computer graphics labs in the world before Microsoft as a company had made any significant investments in computer graphics. Again, you make these investments well ahead of when they're going to be valuable. The work that we did in deep neural networks, that led to the very earliest real time speech-to-speech translation. Back in 2012, I was in China, gave a speech. I demonstrated simultaneous translation from English-speaking to Chinese in my own voice to an audience of thousands of Chinese students. And there were literally students in the audience back then that were crying. They didn't think this technology was ever going to exist.

It was my first million YouTube video thing, million views. Again, we made these investments early on, and then eventually they became important parts of our product strategies, and it also had a huge impact on other people's research that led to where we are today in a lot of areas of AI.

Becky Monk:

I want to talk about that a little bit because I know there's something out there about, I guess, when did you really think that you guys had made it in to the outside world? You were already

making strides inside the company. They were seeing your value inside the company. When do you think that the outside world recognized Microsoft research as a research powerhouse?

Rick Rashid:

So it's interesting you mentioned that because there's a specific incident in time when that happened. So we had built up one of the strongest basic research groups in computer graphics, and I think it was the 1996 SIGGRAPH being the most important computer graphics conference. And the, in that year, more than 25% of all the papers in the conference had a Microsoft Research author. And I can remember this has never been done before, right? This is extremely unusual to have a single research lab have that kind of influence on one of the top conferences. And I remember going into Nathan Mhyrvold's office at the time, and I'm sure we were thinking exactly the same thing. We both had gotten this news and we just looked at each other and just started laughing. It's like we've made it. This was a coming out party for Microsoft Research because people realize, here's this research. It's only been going for five years, and yet they're suddenly having this big impact in this conference. And then you start to see other conferences, other major conferences in other areas having 15, 20, 25% Microsoft Research authors. It's like, who are these guys?

But that was a point in time where it became clear that Microsoft Research had become a significant player in the research world and was also a point where we started to expand in a fairly significant way. So we'd grown fairly steadily over that period, but not huge. We weren't a particularly huge organization. I think we'd gone by 1996, I think we maybe had a hundred researchers. And I remember then going back to the impact we had on the release of Windows 95 and Office and so forth, Bill sent me an email basically saying, okay, you guys are doing great. I want you to triple the size of research. Now. I think Bill may have been thinking more in terms of how product groups work, because you could grow a product group fairly quickly. You can't grow a research group that fast.

It takes way longer. But look, Bill, I can't do it that quickly, but we can grow and we will grow. And that's when we also started making investments outside the United States because there was a limit on the amount of talent we could access just within the United States in any given year. But we started our research lab in Europe, in Cambridge, England. We started a research lab another year later or so in China, in Beijing, Microsoft Research, Asia. And so now we were able to geographically expand and we could bring in many more researchers from new areas of the world. So we could tap into people who really wanted to be in England or in Europe. We could tap into the Chinese young Chinese researchers. And then after that, we eventually started a lab in India. So we started growing, but not just within the us. We grew broadly. And again, that allowed us to bring in a lot more talent than we would if we'd only confined ourselves to the United States.

Becky Monk: How many labs and how many researchers did you end up with when you left the company?

Rick Rashid: Oh gosh. Well, it kind of a little bit depends on how you look at it. I mean, the major research labs were, we had a research lab in Redmond, but we also had one on the East coast in New England. We had a small group, modest group in the Bay Area, so roughly three in the United States. We had the research lab in Beijing and the one in Bangalore and also in Cambridge, England. But we also had an advanced research group in Germany in Egypt. We had a virtual research collaboration in Latin America. So the reach of the organization was fairly significant. And I don't remember, honestly, just don't remember precisely the number of researchers we had in total then, but probably on the order of 1,000 to 1,200, something like that. Certainly one of the larger fundamental basic research groups in computer science. Probably if you think about the very largest organizations and if you want to take into account graduate students, then you would look at places like Carnegie Mellon or

MIT. But Microsoft Research within the industrial world I think was considered to be one of the larger and more influential.

Becky Monk: When you think back on all the different things, how many patents do you have?

Rick Rashid: I don't know, but it's dozens. I've never actually sat down to count, but they're quite a few. And they've played roles in some of Microsoft's legal ventures over the years.

Becky Monk: That's a whole other podcast or something.

Rick Rashid: It sure is. One of the things I told you that I had an amusing story about my last name, and that is a legal story. Interestingly enough, you wouldn't imagine my last name having a legal impact on Microsoft, but it actually did. So again, really early days of Microsoft Research. We only had a few teams, one of which Nathan Mhyrvold had originally helped to recruit some of the key people for, which was our natural language group. Karen Jensen, Steve Richardson were the leaders of that. George Peor. And again, really early days, I mean, we were back in building nine. It was very tiny. We got approached by the legal team. They came to my office. They said they wanted to have a person that ran our natural language group be there. So Karen Jensen showed up, and I knew something was weird when they came to my office, they took a look at me and they had this really big set of grins on their faces, and when people just look at you and they start grinning like that, there's something strange going on. So what had happened was that Microsoft was being sued by a Qatari company named Lamaya, and they were alleging that we had stolen technology from them.

Basically what they were saying is that WAI had gone defunct because of Saddam Hussain, and this company went away for a little bit. Some of their people fled qa, some of them wound up in England, some of them wound up working at Microsoft. QA is back

on its feet, company gets back on its feet. They look around, they say, oh, some of our old people at Microsoft, and they're doing some things related to what they were doing for us. They're probably stealing our technology. I don't think they had really good belief that we really were, but they thought that they could make some money by suing us.

So there was this lawsuit, but one of the things that their lawyers had concocted, and this is just a story lawyers make up stories sometimes, was that that we had this natural language research group and that we were somehow using it to help steal this technology, this natural language technology from this other company, taking advantage of these employees that were now had lost their jobs because of the war, but then had eventually come to work for Microsoft. And they touted the fact that there was this, what they would refer to as Dr. Rashid, clearly an Arab person of some sort running this thing. And so in their minds, they could make a jury believe that this Dr. Rashid was conspiring to help steal their Arabic language technologies. Literally, I get up in federal court in Seattle to give my testimony.

Of course, I don't really have anything to do with any of this stuff. They just made this stuff up. But they tried to sell this idea to the jury. So our lawyers, they put me on the stand and the first thing they say is Dr. Rashid. So they pronounce my name correctly. Say, where were you born, Iowa. So clearly at that point the jury says, wait a minute, Iowa doesn't seem to have a whole lot to do with the Middle East. Do you read, write, or speak Arabic? Now, it just went on that, do any of your friends do this? They just kept nailing one nail in the coffin after another by making it clear to the jury that the other side, the plaintiffs had just made a story up. And this Dr. Rashidhe's from Iowa, he doesn't know anything about any of this stuff. He doesn't look like he would know anything about this. So my joke there is that I saved Microsoft. I think we were being sued for like \$11 million, which was a lot of money back then. I saved

Microsoft \$11 million, and that probably paid for Microsoft Research for the first three or four years later on, one of the lawyers for the plaintiffs had become a lawyer for Microsoft. He and I had just laughed about the whole thing. He said, you know what? As soon as you got up on the stand, we knew we were done. That was the end of it. But that's how the pronunciation of my last name can actually have an impact on the fortunes of Microsoft.

Becky Monk: Oh my gosh. That's probably one of the easiest cases that Microsoft had to battle in court.

Rick Rashid: Like I said, when the lawyers saw me, they had a very big grin on their face. We're winning this one. That's right.

Becky Monk: Thank goodness. Alright, well, let me ask you this. You had your hand in so many innovations for the company and beyond. Do you have any that are your favorite children of research that you can talk about and how they impacted Microsoft and the future of Microsoft?

Rick Rashid: Gosh, well, I worked on quite a few different projects. I think one of the things that I am rather proud of is something I mentioned earlier, which is the work that we eventually did to advance speech-to-speech machine translation. And that's really an example of, so going back to now 2008, so several years beforehand, I'd come up with this feeling like I needed to do something to try to help stimulate people to really push this state of the art. And I thought we should just be doing more, give people license to do really out there kinds of research projects. And so I came up with this notion of the Impossible Things initiative. And the idea was I had this memo and I sent it around. The idea was that it goes back to the Lewis Carroll's "impossible things" before breakfast that we should look at some things that in some sense you could say at that instant in time were impossible, but that could be a stimulus for ideas that could have an impact later on.

And one of the few that I spent some time writing about in that memo was the idea of doing this speech-to-speech translation. And I said, look, every year I go to China as part of our 21st century conference that we'd run for tens of thousands of Chinese students each year. And I get up on stage, I would give my speech, but there would be translators, human translators in the back that would do the translations. What if I could get up there and give my speech and have a computer system do the translation and to do that as I'm speaking so that people could hear me speak in Chinese? And at the time, that really seemed very far from possible the speech recognition systems of those days, just the recognition part had pretty significant errors. And so it seemed like it was going to be really hard to make work. Now I remember, in fact, after I sent that memo out, one of the things I would do is I would have lunch with various teams of researchers in their cafeteria, and I would mostly just, I'd go to the cafeteria, I'd just sit down with groups of people and we'd just talked. That particular day I happened to sit down with the speech recognition team.

I remember one of the young researchers was basically just haranguing me and saying, look, this is not possible. The rate of progress in speech recognition has been just turtle slow for the last 15 to 20 years, and what you're talking about isn't going to happen for more than a decade. And in terms of just the recognition part. And so he just kept going on about why this wasn't going to work. I kept giving suggestions. Well, I'm really, when I speak in public, especially when I speak in a foreign country, I speak very methodically. I'm from Iowa, so I have a fairly nondescript accent, so it's not that difficult to understand me. You have thousands of hours of my speeches already, so you've got a lot of data to work from. You have a pretty good idea of what I'm going to say because in my speeches I often say the same things. I kept giving you reasons why this may not be as hard a task. You could cheat here, you could cheat there, or whatever. I didn't convince him at all, but it felt weird in my mind that here's this young researcher that was basically

saying something was impossible, and here's this old guy basically saying something could be done right. This is the reverse of usually how you expect the relationship to go.

And the teams worked really hard, both in China and the U.S.. They actually worked on this pretty hard. A couple of years after I sent the memo out, they thought they'd gotten things to a point where they could make a go of it, right? They came to my office, I tried this system out, we experimented with it on stage in China, and it just didn't work. I mean, it kind of worked, but the problem was it would make mistakes that seemed really crazy. And I said, look, I don't want to be the Microsoft executive who gets translated into saying some offensive thing in Chinese. This is not going to work for me. And the team was very disappointed, but I said, look, it's not ready. We're not going to do it.

But they kept working on it and they started incorporating new ideas and new technologies. So this is when we began to bring forward the notion of using these deep neural networks and working with people like Geoffrey Hinton and some of his team in trying to adapt deep neural networks to this particular task. And some of that was the actual network technology. A lot of it was actually the technology for training the models because it was painfully slow to train these models. So they had to develop both techniques for using graphics cards of the day to accelerate the training, but also to develop networking technologies so they could get large numbers of machines simultaneously working on the model so they could reduce the training time from six months to weeks, which is an important part of this.

So again, fast forwarding to 2012, I got on-stage attention, well, I should go back earlier that year, they came to my office with their new version, and I knew things were great when we'd accidentally left the system on and we were just chatting and it was transcribing everything we said, and it was incredibly accurate. I said, okay, this

is looking really promising. I went on stage in China and as I said, spoke some amount of my speech in English, it got translated into Chinese and they'd gone the second step of making the Chinese sound like me, using my voice to help modulate it. And it was tremendously influential. It basically proved that you could do this, and I think empowered not just researchers in Microsoft and product teams at Microsoft, but also researchers around the world and product teams around the world to say, oh, you can make this work. We have an idea now that this can happen. And I think that was extremely influential.

Becky Monk: That just has to be so gratifying and so kind a cycle of inspiration to see what you dream up, come to life. That has to be wonderful.

Rick Rashid: It was exciting. I mean, I must admit, even from my perspective, I didn't cry on stage, but there's an energy when you create something new and you're able to demonstrate it for the first time and you can see the audience visibly affected by it. There's an energy that goes with that. And I've luckily had that experience a number of times in the past for things that I've done, but it is, again, very gratifying.

Becky Monk: What do you think Microsoft's legacy is in the world?

Rick Rashid: Well, I think Microsoft is more probably than any other company. I mean, we really popularized the notion that computing was about people. That this goes back to the original mission statement of a computer on everybody's desk. How do you make computing something that empowers people in their normal lives? It's not just something that's about business. It's not just something that's about the science. This is really about what you can do as an individual, whether it lets you learn faster, whether it lets you accomplish some task you couldn't previously accomplish, whether it just entertains you or excites you, informs you. Those are the things that I think if you go back to a lot of the investments

Microsoft made, whether they were investments in their early CD-ROM multimedia PC in a world where a lot of it had to do with entertainment, but also education empowerment, giving people the sense that they could accomplish more than they previously could. It's so funny, I can remember this even goes back to the days of DOS. I can remember going into a, I think it was a Computer City, one of these old computer retailers back in the days that, again, it doesn't exist anymore, but back then it was there and DOS 6 was just coming out. And of course I'd worked on DOS 6 with the compression work that I mentioned

earlier. And I remember coming out of that store and I heard probably a 12-year-old boy just babbling on to his parents about how exciting DOS 6 was and all the new features and all the things that he was going to be able to do with it. And at that point you kind of realize, wow, this is really reached out to everyone that this is something if 12-year-old children are convincing their parents that they should buy a new operating system, then computer science is really come of age. And I think Microsoft is a company that helped, as much as any other company, to make that happen. And we've continued to, I think, push the envelope, whether it's in technologies like the most recent work that we've been doing in AI technologies, trying to bring a completely new set of capabilities to people, whether it's for creation, whether it's for entertainment, whether it's to do their jobs better, whether it's to learn better or learn faster. I think again, that's where we've excelled over the years. It's something that's always made me proud.

Becky Monk: You've always been sort of at that space where you could look in that crystal ball and imagine what was coming. What do you see as the future for Microsoft and for technology?

Rick Rashid: It's funny that you say, I've had this ability to look into the future because honestly, I don't think I necessarily have back in the days of Y2K, I dunno if you remember when Y2K was a big story. I got

called up by a reporter and they were asking me to predict 20 years into the future, what would the future be like? And my reaction was, well, who have you talked to already about this? They'd spoken to sort of a self-described futurist. They spoke to some other scientists and then they spoke to someone who was a self-described warlock. And I said, my money's on the warlock.

I think there's always a danger for people in the field to be confined to their thinking around the field of what they know. And what I've always been excited by is just trying to think of things that are new, just constantly learning about new ideas and new technologies, really trying to push myself to think about what might, is this an idea that might make sense or not? And I can't predict 20 years from now, I can't predict the long-term future. Certainly, I mean, what you're seeing now are things that if you'd asked me five or six years ago, I would've said we would be making strong progress in because we're seeing the fruition now of a lot of ideas and technologies that in the research labs back then. So in that sense, I think you can make some predictions about where you might be going based on where you are.

We're clearly having a tremendous impact on our ability to create, and whether that's artistic creation, whether that's technical creation and engineering, biology, create new medical techniques. If you think about our ability now to tailor treatments to specific diseases, we've just made tremendous strides in that. And if you thought back 10 years ago, people had no idea that you might be able to be here. Now, honestly, I'm alive primarily because of computing technologies that then influence the development of medical treatments because back in 2019, I nearly died of leukemia. I mean, I was in the hospital for a long time. I was in terrible shape. I mean, I needed oxygen. I was getting blood transfusions once a week just to stay alive. I wasn't producing enough of my own blood to be worth a darn.

And luckily, just as I was just about to not be around anymore, a drug combination, which really set a monoclonal antibody in a targeted biologic came to be approved by the FDA. It was used on me and basically within six months cured me. For as much as you can say you're ever cured cancer, I mean anything could always come back. But certainly I don't show any signs today. And in fact, just this summer, I've run a marathon, multiple half marathons, a bunch of trail races. I'm doing fantastic. But those technologies, those things, those new tools for targeting, whether it's cancer or other kinds of diseases, would not exist without the computing technologies that we pioneer Microsoft and in other companies in the field. And I look at that and I say, those are the things that I think are making a big contribution. What will the future hold?

Well, one thing for certain, we will be able to create our own new instances of matter, meaning we will be able to design new materials using the computing technologies, whether it's AI technologies or the coming quantum computing technologies. We'll be able to do that, and that will have an enormous impact on what will be able to do next. Because once we can design materials to our own specifications, we can now start to do incredible things. In some sense, when people talk about what is quantum computing, what's going to be the impact there? They don't, they tend to think about things like, how's it going to impact traditional programming? How is it going to impact cryptography? That's like child's play. Once we have that technology, we should be able to design materials that will then allow us to do even more computing than we could do before that will allow us to tackle diseases we couldn't tackle before, to build, whether it's spaceships, starships, cars, whatever we want to build in ways that we couldn't think of before. And I think that's where sometimes people put their blinders on and they don't really realize how much of an impact, how much of the world is going to change in the next 25 years. If you look, just think about what things were like 25 years ago, the World Wide Web, we just don't have any notion of where the

technology was going to go, and yet it has changed everybody's lives.

So I can't predict the future, but it's going to be great.

Becky Monk: Well, is there anything that you wanted talk about or we should talk about that we haven't yet? I mean there's so much, but is there anything specific that we just didn't get to that you want to make sure we capture?

Rick Rashid: I think we've actually talked a lot about a number of different topics. The thing I always keep coming back to coming is the fundamental value of investing in the future. Sometimes we forget how important it is to invest in fundamental basic research. And by that I don't just mean invest in the technology, whatever that means. It's the people, right? It's creating this steady stream of people that operate at the state of the art in their research areas and are constantly questioning what else could be done? Where are the weak spots in the frontiers of knowledge where if I poke pretty hard, I can suddenly make a breakthrough? And I can remember a reporter coming to my office, and the guy was basically, in some sense, he was trying to poo poo some of the research stuff. And he was saying like speech recognition. He was saying, look, when are we going to get real speech recognition? And I looked at him, I said, have you called your bank recently? Because I think that's a computer that you're talking to.

And yes, in that particular point in time, it was still fairly primitive, but now everybody dictates to their phone. I mean, people are talking to their computers all the time. The latest technologies are so good that it's sometimes hard to distinguish whether you're talking to a person or not. But again, everybody keeps saying, well, we've been promising this idea for a long time, but when it comes, they often don't even notice because the world just changed around them and they didn't realize that it was because of that. And

if you think about what is the value of all the investments that were made over 50, 60 years in speech recognition and natural language understanding, we're living in that world now. We can talk to our computers and they actually do a pretty darn good job of understanding us, and it's just going to keep getting better, and we're going to be able to have that kind of natural interaction with our computing devices that we dreamed about 50, 60 years ago back when I was a young guy.

Becky Monk:

When you think about making sure that that investment in peer research continues, what is the advice you give to young people who are coming up in the sciences so that they can keep that open mind so they can keep poking at the edge of science? What are the things that you tell them?

Rick Rashid:

There are several. The simplest is just don't limit yourself. And I think the biggest problems I see sometimes with young researchers is that they get caught up in the notion that they can't do something. And you have to keep trying. You can't limit yourself and say, oh, this is too hard. I can't do it. Where you stop, right? If you get to a point where you feel like things have gotten to a point, you just can't push on them anymore, then you're probably should do something else. But the other part of it, which is probably more subtle, but it's advice I always give young researchers is don't stop learning. Because so many research ideas come from learning about new things and applying them to things you already know.

If I constantly want to learn, whether it's learn new ways of programming, learn new ways of doing things, just learn new physical things, learn how to dance, learn how to fly an airplane, learn how to run a marathon, whatever, just anything that's going to continue to stimulate your mind and help you make new connections, I think that's where a lot of really exciting research comes from. It's in the ideas that cross over between disciplines or the ideas that seem to come out of nowhere when you make this

connection between A and B where they didn't seem related, but suddenly you discover that they are. I remember one of our researchers, in theory, Jennifer Chayes, she did her original work in statistical physics, and one of the things that she worked on early on was looking at the relationship between things in statistical physics. So basically the phase transitions in matter and large graph theory and the complexity theory in computer science.

And she and a number of other people that she collaborated with and people that other research as well over a period of time gradually realized that those were all related problems. And so the mathematics that had been developed in one area could suddenly now be used in these other areas to tackle problems there that people had gotten stuck on. And I think that's where things can be really exciting is that to suddenly realize that these otherwise unrelated problems or unrelated things can suddenly be the same if you think about them the right way. And so that's why I keep telling young researchers, don't stop learning something new. Go and talk to people in other areas, learn what they're doing, think about how that might, is there anything there that makes any sense relative to the things that you're doing? Because a lot of times these ideas do have an impact. So those are two bits of advice that I usually give young researchers.

Becky Monk: Fantastic. Last question. What do you think your legacy with Microsoft is?

Rick Rashid: Probably the single biggest legacy is Microsoft Research. I mean, in some sense you create an organization, you hope that it continues on with the same set of values and the same perspectives. And certainly one of the things I find gratifying is when I meet with now older, but originally very young researchers that I work with who will tell me, look, you had a tremendous impact on my career. You changed the way I thought about what I was doing. And then they in turn have an impact on the young researchers that they work

with. And it's that sort of generational passing of perspective and knowledge that you hope will continue and have an influence. To be honest, I think the Microsoft we have today exists because of the investments we made in basic research. So you could argue in some sense, part of that legacy is the company itself as it exists today, and its willingness to embrace change. I think the early founders, the early executives, they were, I think, important in helping to make those decisions and to embrace change. But it's something you have to continue to do, and I think having a research organization is one way to help push you to do that.

Becky Monk:                    Fantastic. Thank you so, so much.