

[MUSIC PLAYING]

Good afternoon, ladies and gentlemen, and welcome to the second [INAUDIBLE] presentation by the Distinguished Speakers Series [INAUDIBLE]. We are honored to have with us today Steve Jobs, the president and chief executive officer of NeXT Computer Corporation. Steve has consistently been one of the computer industry's foremost entrepreneurs. Along with partner Steve Wozniak, their entrance into the industry was by the traditional entrepreneurial route of building computers in a garage.

[INAUDIBLE] auspicious start, however, [INAUDIBLE] the creation of Apple, a formidable challenge to IBM in the personal computer market. A true visionary, Steve was known at an early age for his belief that a PC could be put on every desk and made simple enough for use. Educationally, he had the right credentials to make it big in the computer industry, because he dropped out of Reed College.

[LAUGHTER]

[INAUDIBLE] and the growing corporate culture diverged, both eventually left Apple to establish their entrepreneurial roots. In 1985, Steve founded another computer company, NeXT Corporation, to design a revolutionary computer called the Cube. Targeting the high end workstation market, NeXT has found the going tough against [INAUDIBLE] competitors, such as Sun Microsystems.

However, with 1991 revenues hitting \$127 million, business has grown substantially. With the backing of Ross Perot, NeXT shouldn't be short of funds in the future. Or in a more unlikely scenario, a friendly year in the White House. The current plans for NeXT computer involve a public offering, possibly in the next 14 months. We hope to hear today just what a good investment that should be.

Steve has consistently been recognized for his contribution to America's largest high tech industry. Most recently, he was appointed to the president's Export Council last August, with the role to advise on government policies and programs that affect trade performance. Outside of work, Steve is married to Laurene, previously an MBA student at Stanford University, and also a sister of one of our classmates, Brad Powell. The couple met when Steve gave a similar speech to this afternoon's at Stanford, days before the Bay Area earthquake. Let's hope that history doesn't repeat itself.

[LAUGHTER]

Ladies and gentlemen, we are extremely grateful for Steve to giving up his time to be with us this afternoon. Could you please give a warm welcome to our distinguished guest, Steve Jobs.

[APPLAUSE]

Thank you. Hi. I guess we get to spend an hour or so together today. And most of the time, I wanted to spend just talking about what you want to talk and answering some questions. But I thought if you wanted, I'd take about 10 minutes or 15 minutes upfront and tell you what we're doing at NeXT and why the world might need another computer company. Is that something you guys think about these things? OK.

I thought I'd tell you about some of our mistakes. Maybe that would be more useful. We have a lot of scar tissue. There is a really interesting book that was written by a guy named Paul Strassmann. And Paul has one of the more interesting jobs in the planet. He's the Chief Information Officer-- CIO of a very large organization called the Pentagon. And they really understand software there. I had a conversation with him not too long ago, and he said, the lesson from the Gulf War was that the best software will win the war.

And so they're trying to do a lot of work in the software area. He wrote a book, though, before he got this job called *The Business Value of Computers*. It's rather thick, and it's not good bedtime reading. But you can plow through it, and there's some incredible stuff in it. And he asked two questions in particular. One was, he surveyed a bunch of companies from not very successful all the way up through really successful. And there's somebody taking notes here.

[LAUGHTER]

He asked how much they spent on information technology as a percentage of revenues. And he got a very counter-intuitive answer, right? You'd think that either the really successful companies would either spend more or less than the not-successful companies, depending on your theory. But it was exactly the same. They all spent about 2% of revenues on information technology.

And he found this curious, and so he asked another question. How did they spend their money? And he found out that the really successful ones-- actually, let's start with the not-so-successful ones. As success increases and dollars increase, he found out that the not-so-successful ones spent the majority of their money on management productivity, and the more successful ones spent the majority of their money on operational productivity applications.

Now, this was not very pleasant for me to read, because I spent the first 10 years of my life on management productivity, which was PCs. PCs and Macs never attacked operational productivity. They just attacked management productivity. Why is that? Because you can't go down to your local computer store and buy an app that will help you do stock trading, or will help you run a hospital, or will help you in whatever operational part of your business you want to automate. Unless you're a very, very small business, then you can run some accounting packages. But other than that, if you were a medium-sized or large business, these things never attacked operational productivity.

So we zoom out and we say, how have people attacked operational productivity with information technology? Well, in the '60s, they bought a mainframe, and they got some terminals and a bunch of COBOL programmers, and they wrote a few apps. And most of them were kind of back-room apps. And it sort of worked for the very few that could afford to do this.

In the '70s, they got a mainframe and some terminals, and they did the same thing. And a few of them got a few mini computers and terminals and tried to do it a little cheaper. In the '80s, nothing changed. Mainframe and terminals, minis and terminals. Until maybe about two or three years ago. What happened two or three years ago was that the front office started to realize that they needed operational apps so bad that they couldn't depend on the MIS folks anymore.

They started taking life into their own hands and sometimes working with the MIS folks to start downsizing and getting some servers and running some industry-standard databases like Sybase or Oracle in the servers, and making a little local area network, and getting maybe some Sun Workstations, and spending about two years writing some mission-critical operational applications. Like trading apps for Wall Street, perfect example.

And it kind of worked. And the reason that they needed to do this was because more and more, they were discovering that things like new products required a custom operational application. An example, if you're in financial services and you come up with a new product, it's only three things. It's an idea, it's a sales force, and it's a custom app to bang on databases to make the product real, to do the mortgage swaps or whatever it is you want to do. Without the app, you don't have a product.

And so there has been an increasing buildup of demand from the front parts of corporations to create more and more and more of these operational applications. And I think it's going to get to the point where this becomes fairly clear that this is the next big revolution in desktop computing, is to attack the operational productivity. And as we start to re-engineer the way we do things, to automate a lot of this in custom applications.

Sounds a little strange now, to most people. Sounds like desktop publishing in 1985. Nobody knew what it was, everybody thought it was kind of a strange vertical thing over there. But my guess is it's pretty horizontal. And we're attacking vertical markets now that know they want this. And it's going extremely well. Sun is the only company that's really had any success at this, and we're knocking them out of the box. Because we came up with the software called NeXTSTEP which lets you build apps five to 10 times faster than anything anyone's ever seen.

And after you build them, they're deployable and usable by mere mortals, because it's really easy to use, this computer. And you can interoperate your custom apps seamlessly with a bunch of off-the-shelf productivity apps. So we go to these companies that use Suns and take two years to write their apps-- or are thinking about using Suns, and they can write their apps in about 90 days on a NeXT. Now, if you're on Wall Street and you can create a new product in 90 days versus your competitor in two years, that's eight new products you can field for their every one. And you can start to see the competitive advantage that can be created this way.

Now, we had no idea that we were any good at this when we started NeXT. A lot of times you don't know what your competitive advantage is when you launch a new product. Let me give you historical example. When we created-- how many of you guys use Macs? Anybody? Good.

[LAUGHTER]

How many of you have seen a NeXT? Oh, how many of you use a NeXT? Oh, that's not so bad. We'd like to change that ratio a little bit.

[LAUGHTER]

We're on the right track. When we did the Macintosh, we never anticipated desktop publishing when we created the Mac. Sounds funny, because that turned out to be the Mac's compelling advantage, right? The thing that it did not one and a half or two times better than everything else, but four or five times better than anything else, where you had to have one. We never anticipated it. We anticipated bitmap displays and laser printers, but we never thought about page maker, that whole industry really coming down to the desktop. Maybe we weren't smart enough.

But we were smart enough to see it start to happen nine to 12 months later. And we changed our entire marketing and business strategy to focus on desktop publishing, and it became the Trojan horse that eventually got the Mac into corporate America, where it could show its owners all the other wonderful things it could do.

Likewise, when we created NeXTSTEP, this revolutionary object-oriented software that we have, our target customer coming from the PC world, where shrinkwrapped apps were king, was Lotus and Adobe and WordPerfect and all the shrinkwrapped apps developers. And the purpose was to let them create their apps five to 10 times faster for these shrinkwrapped apps. And it worked. We have a ton of shrinkwrapped apps, now. Best of breed in almost every category.

But it wasn't until early in '91, early last year, a little over a year ago, that some really big companies came to us and said, you don't understand what you've got. The same software that allows Lotus to create their apps five to 10 times faster is letting us build our in-house mission-critical apps five to 10 times faster. And this is the biggest problem we've had.

This is a huge problem for every big company, and almost all medium-sized companies, and you have a solution in your hands, and you dummies don't even know it. And it took them about three months before we finally heard it. And then last summer, we changed our whole sales and marketing strategy around to focus on that. And it's taken off like a rocket.

And we grew about 4x last year, and probably grow about 2x this year. And our customer list is now very, very strong and growing like crazy. We just got back from spending a few days in DC and in New York. And we're talking to customers we only dreamed of talking to a year ago. So that's what we do. And our arch enemy Sun, they want to kill us. Which is good. They should try to do that as soon as possible, because the sooner they do it, the cheaper it will be for them.

[LAUGHTER]

I think it's gone past the point where it's possible. And the greatest thing is, hardware churns every 18 months. It's pretty impossible to get a sustainable competitive advantage from hardware. If you're lucky, you can make something one and a half or two times as good as your competitor, which probably isn't enough to be quite a competitive advantage. And it only lasts for six months.

But software seems to take a lot longer for people to catch up with. I watched Microsoft take eight or nine years to catch up with the Mac, and it's arguable whether they've even caught up. It takes a long time. And we think that the soonest we're going to have a true competitor is probably four to five years. So we've got that amount of time to grow ourselves a \$1 to \$2 billion company so that we can compete with them on scale.

See, today we can't compete with them on scale. We never have as many salespeople as they do, as Sun does. We don't have the ad budgets that they do. So we've got to have a better product. And I hope we always have a better product, and I think we can. But I'd also like to be able to at least give them a run for their money on scale. So we've got the next three to four years to run really fast, so that by the time they even get close to having a competitive product, we're at a large enough scale to where we can start to compete with them.

And that's what we're doing with our lives right now, spending a lot of time with customers, spending a lot of time making NeXTSTEP better, and that kind of thing. So that's the strategic basis of what we do. Does that make any sense to you? Have you run across the concept of sort of operational custom applications at all?

I mean, most of you have come from companies where you've had work experience, right? And you've all done that? So do you have this problem in the companies you come from, of a lot of pressure to write these operational custom applications and hardly anything coming out of the spigot to satisfy this thirst? How many of you from Wall Street? Good.

[LAUGHTER]

Good How many of you from manufacturing companies? Excellent. Where are the rest of you from?

[LAUGHTER]

Consulting.

How many from consulting? Oh, that's bad.

[LAUGHTER]

A mind is too important to waste.

[LAUGHTER]

You should do something.

Why is that bad? A consultant can come into a company and use your system, and basically build their applications in predictably short amounts of time, and show them a working product.

The only consultants I've seen that I think are truly useful are the ones that help us sell our computers. No seriously, I don't think there's anything inherently evil in consulting.

[LAUGHTER]

I think that without owning something, over an extended period of time-- like, a few years-- where one has a chance to take responsibility for one's recommendations, where one has to see one's recommendations through all action stages and accumulate scar tissue for the mistakes and pick oneself up off the ground and dust oneself off, one learns a fraction of what one can.

Coming in and making recommendations and not owning the results, not owning the implementation, I think is a fraction of the value, and a fraction of the opportunity to learn and get better. And so you do get a broad cut at companies, but it's very thin. It's like a picture of a-- I'm a vegetarian, so I won't use steak. But it's like a picture of a banana.

[LAUGHTER]

You might get a very accurate picture, but it's only two dimensional. And without the experience of actually doing it, you never get three dimensional. So you might have a lot of pictures on your walls. You can show it off to your friends. You can say look, I've worked in bananas, I've worked in peaches, I've worked in grapes. But you never really taste it. And that's what I think.

[APPLAUSE]

You're also a variable expense. And in hard times, you find yourself. You find yourself variable, right?

If it's the software that's going to make or break your company, how come you're putting out on platform. Why don't you put it on Sun's? They have a much larger base.

Right, very good question. I'm going to generalize your question. Why don't we just become a software company, right? That's a very good question. It's a subtle question. I'm going to try to go through some several things, and I'm sorry if I jump around. We got a lot of requests from customers last year that they would love to see NeXTSTEP on other platforms, and primarily, Intel-based platforms like the 486.

So we decided to do just that. And we have ported NeXTSTEP to the 486, and we're finishing it now. And it will ship in the September, October timeframe. And it's exactly the same stuff we run in our own computer. Same app, same user interface, same training, same development environment. And we're going to sell it for \$9.95, and we're OEMing it to a bunch of companies whose names you'll recognize quite easily, and OEMing it to them at a much cheaper price.

And everybody's coming out of the woodwork to help us. We're getting help from Novell. We're getting help from all the developers. Intel's really helping us. And they really want us to succeed. Why is that? Right. They all want to make sure that there's a choice out there, and they're all really scared about Microsoft. And they see NeXTSTEP as the only thing on the horizon that can challenge Microsoft in system software for the next several years. So we're enjoying a lot of help, and boy we need it, so that's good.

Now, we've also had a lot of requests from companies to port NeXTSTEP to other platforms. And we're talking to some of those companies right now. Now, we've got a lot of requests from Sun customers the port NeXTSTEP to Sun. So a lot of them are saying, look, we may not want to buy them anymore, but we already bought 500, and we don't want to throw them in the bay. So can we put your software in them? Because Sun's falling behind in software. Now, Sun says they'd rather stick needles in their eyes than help us do this.

[LAUGHTER]

That's a quote. And so we're evaluating right now, which will be worse for Sun, if we port it or if we don't port it? And since we're fairly customer-driven, we'll probably end up doing what the customers ask us to do, because we want to make them happy. Now, this leads one into the question, should we just be a software company?

And we think the answer is no. We think we should be a software company and a hardware company. In making the decision to put NeXTSTEP on more than our own platforms, we clearly decided that we will sell less than 100% of the NeXTSTEP hardware. However, we think that the marketplace will grow, and we will sell more absolute hardware.

And secondly, the charter of our hardware division is to make the best NeXTSTEP hardware. Might not be the cheapest, might not be this, might not be that. But we think all in all, we can make the best stuff. And I would love at nothing better than if someday we only sold 20% or 25% of the NeXTSTEP hardware. But I still think that's a billion-dollar-plus hardware business. And I'll get into one other hardware-related reason in a minute.

There are some things I can't talk about here. In addition to that, if you look at how we sell our computers right now, we have a sales force in the US of about 130 professionals in the field out selling NeXT computers. They spend 90% of their time selling NeXTSTEP software, and then 10% of their time selling the hardware.

In other words, if they can get the customer to buy into NeXTSTEP, then they're going to sell the hardware, because right now we have the only hardware it runs on. So they are out there selling NeXTSTEP right now. And this is what is required to launch a new innovative product. The current distribution channels for the computer industry over the last several years have lost their ability to create demand.

They can fulfill demand, but they can't create it. If a new product comes out, you're lucky if you can find somebody at the computer store that even knows how to demo it. So the more innovative the product is, the more revolutionary it is and not just an incremental improvement, the more you're stuck. Because the existing channel is only fulfilling demand. Matter of fact, it's getting so bad, that it's getting wiped out, because there are more efficient channels to fulfill demand, like the telephone and Federal Express. So we're seeing the channel become condensed on its way to I think just telebusiness.

So how does one bring innovation to the marketplace? We believe the only way we know how to do it right now is with the direct sales force, out there in front of customers showing them the products in the environment of their own problems, and discussing how those problems can be mated with these solutions.

A software-only company could never afford to field a direct sales force. With average selling prices of \$500 a software package, you could never afford 130 professionals in the field. With an average selling price of \$5,000, you can. And that's why I don't think we're going to see any more systems software companies succeed. I don't think it's possible to fund the efforts to educate the market about a revolutionary product with ASPs that low.

And if it's not a revolutionary product, I don't think the company can succeed. So our strategy has been that we've got to be a hardware company in order to make our software business succeed. And we think we can do really well at both of them. I know that's a long answer, but it's a complex problem too. Yes?

[INAUDIBLE]

Sure. If you ask us who our competitors are, we'll say really three things. One is Sun's Solaris software, the other is Microsoft, and the third is Taligent. Let's take them in that order. Sun for a while had a software value added, because they had the best Unix in the marketplace. But the market's moved way beyond that, and unfortunately, Sun hasn't. So their software's falling further and further behind, and while we take them very seriously, we don't think Solaris is going to be much competition. It doesn't have an object in it, and it's pretty much what they have today.

Microsoft is doing NT, which is their sort of second attempt that a Unix wannabe. And that's great, and I think it will be better than the last one, which was OS/2. But fundamentally, it's just an operating system. It's better plumbing for Windows. That's a good way of thinking of it. Unfortunately, you're still stuck with Windows in all of its glory, including the worst development environment that's ever been invented. And so we don't think that this is really going to present a challenge to what we're going after, which are these mission-critical custom apps, because the development environment is horrendous, it's not object-oriented, and even with the better plumbing, we think it will be widely rejected for what we do.

In terms of Taligent, Taligent represents the first true competitor that we could have. They're going to ship a product around 1995, if they execute to their plans. And I think if they do execute to their plans and work really hard, by about the mid '90s, they'll have roughly what we have today. And that's not meant as a joke. It still means we have to run very hard, because they will have a lot of resources at their disposal.

I think there's a lot of questions as to whether they will ever ship a product. I think they're a few years away from having something running, unquote. But we take them very seriously. Now, they've helped us enormously, because they've blessed object-oriented programming. And right now, we're the only folks that have it, and will be for the next three or four years. So if we can't compete with Taligent, it's probably because we've shot ourselves in the foot. You couldn't ask for something better. IBM and Apple saying, these guys are right on track, and come back in five years and we'll have it too. It's great. And so the ball's in our court. Yes, in the back.

You describe NeXTSTEP as an environment and an operating system. And what I hear you talk about is a great way to develop an application. I don't develop applications. I use other operating systems. Do you see a fracture of the operating system market into a specialized niche that you're describing, and then the general market for those of us who don't develop apps? Or are you saying that you can then bridge from an environment that allows you to develop apps into an operating system that'll be like Windows or MS DOS on a lot of machines?

Well, let's get our terminology straight. We look at NeXTSTEP as an operating environment. Much more than an operating system. Unix, which is our operating system, is 10% of NeXTSTEP. So in these modern operating environments, when you develop a custom app in NeXTSTEP, it doesn't run on the Mac or on Windows. It can't. You need all the objects that come with NeXTSTEP to make it work.

And let me go into a little detail there. How many of you are technical here? A lot, OK, great. Well, we've discovered something. You don't write code any faster in NeXTSTEP than you do in any other operating and development environment that we know of. However, to do a particular app on average, you write about 20% of the code that you do in any other development environment we know of. About 20% of the code than in Sun or in anything else.

So the code that's the fastest to write, the code that's the easiest to maintain, and the code that never breaks, is the code you don't write. So that's our strategy, write a lot less code. And the way we do this is we enable the developer to use a lot of objects that others have written. We ship six years worth of objects with NeXTSTEP.

You can create your own objects for your own company, and then reuse them around your developers. And there's now independent third party companies not selling system software, not selling applications, but selling objects. There's not quite 10 of them, and it's starting. And I think it's going to be a very big thing.

So in order to deliver these apps that are created on the NeXTSTEP, you have to have NeXTSTEP so they can run on top and take advantage of this rich community of objects. And that's why we're porting NeXTSTEP to the 486. That's why you'll see NeXTSTEP running on several hardware platforms. And will there be a fracturing, if you will? Sure, to some extent. And I tend to look at it as a transition. Was there a fracturing when Mac came out? Yes. And there's been more of a transition, as people either move to Mac or in the case of Windows, adopt what's good about the Mac.

Same thing's going to happen here. We believe very strongly that the benefits from these object-oriented environments-- not only just rapid development, but a much richer user environment. If you use a NeXT, it's a lot nicer than a Mac or a PC, even if you never developed an app. Other environments will absorb some of those breakthroughs, and more and more people will use NeXTSTEP. And things will tend to balance themselves out over time. But there will be a transition. And our goal is to make sure we're part of it. Yeah.

You spoke earlier that your competitive advantage, you felt lied in the fact that these companies today were needing development on the operational basis, as opposed to the management. Many companies, though-- or some, perhaps-- outsource their development. How does this change you business strategy?

Good question. If our business strategy says that we do a lot of things well, but the tip of our arrow, or our Trojan horse, which is getting us into these large and medium-sized accounts is our custom application development ability, then our growth is going to be paced among other things by the development community available to these companies to develop these apps.

Now, even though we have shrunk the development time down to a fraction of what it was, still without developers out there, we're not going to win. Fortunately, most companies have really increased the staffs of good people in their IS departments dramatically in the last several years-- last four or five years. There's now really bright people who know something about computer science in IS departments. And most of the industries we talk to, whether they're healthcare, financial services, even law enforcement, places like that have on-site developers in their IS teams.

If not, they're starting to become a lot of vars, and people out there that we're using in LA or in New York where I just was, we probably have about 10 independent third party companies helping parts of Wall Street develop things. So I think that's not going to be a stumbling block. My personal opinion is the number of people in applications development is shrinking. It exploded in the '80s with PCs, but now that industry is consolidating. It's consolidating down to very few companies. And there's a lot of excess people out there, that I think are starting to get channeled into some of these other areas.

So far, it hasn't been a problem. But if you want to do that, please call us and tell us, because we always need more. So what are you guys all going to do when you get out of here? You going to go back to your companies? How many of you are going to go back? How many aren't going to go back? What are you going to do?

[INAUDIBLE]

[LAUGHTER]

Well yeah, I think we always have jobs for really smart, technical people. You know, technically-based people. And you guys are getting a great business education here. Yeah, we'd love to talk to you about it. Yes, sir.

Well, I don't want to step on any toes here.

Go ahead.

But where do you Apple would be had you not left it? And specifically, could it have come out with something as revolutionary in the tech industry?

Well, these are deep questions.

[LAUGHTER]

I'll tell you, I've obviously thought about this a lot. And I don't want to get into it too much. But I will say that I think everybody lost. I think I lost. And I wanted to spend my life there. I think Apple lost. I think customers lost. And having said all that, so what? You go on. It's not as bad as a lot of things. Not as bad as losing your arm.

So people go on, and companies go on. And I think Apple-- I'm very happy every time Apple ships a Mac. It makes me very, very happy. I think the PowerBooks are decent products. I like them. But Apple has been struggling the last few years. They've been having a real struggle with who they want to be.

And this is nothing new. We always had that. That was part of what kept Apple alive, I think. And there were two camps within Apple. Camp one wanted to be the next serious computer company, and camp two wanted to sort of be the Sony of computers. And that struggling I think was somewhat tearing Apple apart.

And fortunately, the Sony guys have won. They've kind of decided to go be the Sony of computers. And so the PowerBooks are pretty good, but the Quadras suck wind right now, the high end stuff. And they're basically not putting a lot of resources onto the power users on desktops, and they've put most of their best people now on the portables and on consumer products that they're going to be coming out with.

And I think they'll do very well at that. Now, there's a problem in there, in that if you look at the consumer products that sell over a million a year, you can count them on a few hands. It turns out, there's not-- I mean, consumer electronics products, not like toothbrushes. But electronics products. And so let's assume they have one of those or two of those. Let's say they have a product that sells two million a year at a \$500 ASP to them.

So it sells to the consumer at \$795 or something like that. That's what, a billion dollars, right? two million a year, \$500 apiece. It's a billion dollars. They still have to get the other \$7 billion of revenue somewhere. So it's going to be an interesting transition as the ASPs get lower. The volumes have to get much, much higher, and you run into some scale issues when you look out at the consumer electronics industry.

Not that it's not possible. And I think it's going to be really interesting to watch. And we also see a lot of not quite that clarity of movement, but some movement in the Windows and PC world again towards taking what they have now and making it portable. But not giving the top third of the market more power.

You know, System 7 on the Mac was supposed to be the second coming. And it wasn't. It turned out to be an incremental improvement. OS/2 was supposed to take us into new dimensions, and it turned out to be a complete failure. And they're going to throw another \$100 million after it just to make sure you all know about it.

[LAUGHTER]

And so there is a third of those desktops out there that the users and the people that want to deploy more sophisticated apps are-- they're in the desert wanting something to drink. And I think there's a tremendous opportunity to give them some solutions.

Now, that doesn't mean that Apple is not going to be successful. It just means they're going to go off in another direction. Who knows what would have happened had all this not happened. Yes, up in the corner?

You've talked a lot about making products. I wonder if you could talk about the management at NeXT, and if you see changing at all, as you hope, go from \$120 million company to a \$1 billion company.

Sure. Yeah, we're-- we've done a lot in the last year. The most significant things we did were, one, we hired this guy right here. Mike Slade is our VP of Marketing. He spent-- you can stand up.

[INAUDIBLE]

No.

[LAUGHTER]

And he's cheap, too.

[LAUGHTER]

And Mike spent seven or eight years at Microsoft. And I originally met him, he was the Product Manager for Excel when it first came out, and ended up running big pieces of marketing for Microsoft. So Mike runs all the marketing at NeXT.

And Mike came in to NeXT just about the time when we were finally hearing what these big companies were saying about mission critical custom apps. And so we've worked very closely together to redefine the marketing strategy of the company. And Mike is doing just a superb job at that.

The second big thing we did was, about three months ago, a little over three months ago, we consolidated our hardware design engineering, our manufacturing, our worldwide distribution, and hardware service all into one chunk called the hardware division. So from cradle to grave, they have hardware responsibility. And we're in the process of forming the software division right now, so that we're all still under the same roof, but we have clarity about the fact that we've got to make two businesses successful.

The third thing that we did was-- and we have a really great person running that, Rich Page, who is one of the founders of NeXT. And he's doing a great job running that. We hired a new CFO recently, Marcel Gani.

He's from Intel, spent 12 years at Intel doing some pretty interesting jobs. He ran all of finance for Europe. He ran all of internal audit for the Board. He ran all of manufacturing, planning, and scheduling for a while, and things like that. He's a pretty good guy, went to Cyprus for a year and a half, and then came to NeXT. And he's great.

And then recently, as a matter of fact, about last week, I hired a Chief Operating Officer, a guy named Peter van Cuylenburg. And Peter is someone I first met when we were looking for an Executive Vice President to the Sales and Marketing.

And he turned me down for the job. And we subsequently decided not to hire that position. And I've been chasing him for the last year and a half almost. He's really good. He spent a lot of time at TI as [? run ?] Europe, and about three years ago, went to a company called Mercury Communications in England-- they're the MCI of the UK-- and turned that company around to about \$2 billion, and then was promoted to run half of Cable &-- or half of Mercury's parent company, Cable & Wireless, and was running-- they're about \$6 billion.

And he went there when he thought that communications and computing were going to, kind of, come together, but never happened. And he figured out that his real love is in the computer business. He's very, very good, operationally, so that's just happened last week.

It seems like all the good people I really want to hire, it seems take me a year to hire them. Them And it's always been that way, even at Apple. Some of the best technical people, or whoever, it always seemed to take me, like, a year to pry them out of HP, or wherever, and took me over a year to hire-- I think Mike has the award. You're about a year and a half.

And they're all worth it. What happens is, I usually meet somebody that is really good-- I think is very, very good. And you can't get them. And then you go try to find other people. And nobody measures up.

You know, when you meet somebody that good, just, you always compare them to this one person. And you know you're going to be settling for second best if you compromise. And I've always found it best not to compromise, and just keep chipping away.

So I think we're doing quite a bit. It feels like we're running the company a lot better now than we ever have. We've definitely made our share of mistakes. Yeah?

What technological advances do you see coming over the next five to 10 years? and how is NeXTSTEP structured to take advantage of those?

I'll give you a global answer. Then we can descend into some details. I believe that you can use the concept of technology windows opening, and then eventually closing. And what I mean by that is, enough technology, usually from fairly diverse places, comes together, and makes something that's a quantum leap forward possible.

And it doesn't come out of nowhere. If you poke around the labs, and you hang around the Media Lab here at MIT and other places, you can, kind of, get a feel for some of those things. And usually, they're not quite possible. But, all of a sudden, you start to sense things coming together, and the planets lining up, to where this is now possible, or barely possible. And a window opens up.

And it usually takes around-- my experience anyway, my life has been, it takes around five years to create a commercial product that takes advantage of that technical window opening up.

Sometimes you start before the window is quite open. And you can't get through it. And you push it up. And you push it up. Sometimes it just takes a lot of work, took that long with the Apple II, took that long with the Mac. You know, it took a Lisa along the way, \$100 million. It takes a while. It's expensive to push those windows open.

And in our case, you know, our first product failed. We came out with this cube. And we sold 10,000 of them. Why? Because we weren't quite there yet. And we made some mistakes along the way. And we had to course correct.

You know, Macintosh was a course correction off the Lisa. So with Apple II and III, we did it in reverse. It takes around five years, or some number of years like that, to realize that window opening. And then it seems to take about another five years to really exploit it in the marketplace.

And let me give you some examples. from my life. Apple II lasted 15 years, 15 years. The hardware churned. But basically, it was the same for 15 years. DOS-- you know, DOS, just passed 10 years. I don't think anyone would disagree that it's going to easily last another five, right-- unfortunately.

And Mac, you know, Mac is eight years old, right? No question it's going to last another four or five years, right? These things are hard. They don't last because it's convenient, or even because it's economic. They last because they're really-- this is hard stuff to do.

And so when we are pushing that window open, I think with our current generation of products, we finally got the window open. After six years, it's open. We've got an extremely elegant implementation. And we've got five years of work to do to exploit it in the marketplace.

You know, we'll peak in five years. Five years, we'll all sit around, and say, OK, it's time to get started on the next thing. It's time to get going on the next thing-- maybe four years from now. But we've got a lot of work ahead of us just to move this thing out, and educate the market, and continue to refine it based on market feedback.

So everything I know about technology windows that are open, or just about open, is in NeXTSTEP, or we're working on it in the labs. And these things generally don't come along independently. They, kind of-- clumps of them come together, has been my experience.

So the things that aren't in there right now that I can talk about, there's some video stuff that's really interesting, that's going to be integrated in. There some security stuff that's really fascinating that's integrating in. But most of the core technologies in there-- products are getting smaller and portable. Products are getting much, much faster. But these things are well known.

You know, the products that we can put on-- that we can give to you in the next year or two are going to be running at speeds that I find hard to believe. And I've been doing this for a while. So I think the windows open.

And I think object-oriented technology is the biggest technical breakthrough I have seen since the early '80s, with the graphical user interfaces. And I think it's bigger, actually. It will prove to be bigger over time. Yes?

I have a question. If the value [INAUDIBLE] and things like operational applications, and NeXTSTEP is a tool to help you build those, where do you draw the line in capturing the most value between actually developing applications, something like Word or Excel, and becoming a tool builder to support those people who may capture a little bit more of the value? Where do you want to draw the line there?

If you're a software company?

If you're a software company.

Well, Microsoft has made it easy.

Repeat the question.

Yeah, oh, sorry, the question is, if you're a software company, should you develop apps, or should you develop objects and tools? And see, the software industry, with Microsoft, and Lotus, and WordPerfect made it real easy. Unless you've got \$20 million or \$30 million burning a hole in your pocket so you can go out and hire a few hundred people to develop what it takes to develop one of these polished shrink-wrapped apps on Mac or on Windows, then you don't have much of a choice. There's not a real opportunity there to-- assuming you had the best spreadsheet today, you could easily spend \$50 million marketing it before you'd be breaking even selling it, because of just how expensive it is to market a product today.

But that's only for existing spreadsheets. If you're trying to develop a next-level generation of it, a different type of product.

Well, assume that you have a breakthrough spreadsheet. Again, on mainstream platforms, it will take \$50 million to just rise above the noise level and market it. So what the brightest people I know of today are doing is they're writing objects. They're writing hunks of things that other developers are going to use to build apps. And they're going where everybody isn't. And that's, I think, going to be the next new thing. Yeah?

You mentioned that a possible problem for Macs is the lack of developers and consultants out there who recommend it. Based on that view, what's the NeXT view of how to market itself to academic institutions?

I think your question is, what is our philosophy on marketing our products to higher education?

Mhm.

Well, we started off selling only to higher education, which arguably, was a mistake. And But we've done really well there. And our hearts are there. And we sold a lot of Macintoshes there too, when we were at Apple.

And I think it's a no-brainer. I think you take your products. You discount them as heavily as you can. You sell them into higher education. Higher education is a wonderful place to give you great feedback about how to make your products better, and what's wrong with them. And it's a great place to educate bright people that you can hire, and that your customers can hire when they graduate, so we do exactly that.

We have-- I think were sold on about 350 campuses in the United States. We are clearly the number one selling workstation across the US. And we're the number two selling computer of any type at campuses, like here MIT or Stanford, right behind the Macintosh.

We sell more computers at MIT than PCs through the institutional resale engines. So we do as much as we can. And I think it's been pretty effective. What do you think?

I mean, for example, [INAUDIBLE] is going away. Like, the support [INAUDIBLE] is going.

Right.

I mean, I'm talking about, not just selling computers. I'm talking about establishing a network of institutionalized system [INAUDIBLE].

Well, what's happening is-- see, the Project Athenas didn't succeed for a reason. They didn't succeed, because they had a lot of good ideas as a research project, but the people necessary to commercialize those things and make them into real products aren't necessarily the same people that are going to pioneer the ideas at a university like MIT.

And so those projects never quite get baked. The recipes, kind of, developed. And you make a few samples. And it's pretty good, but the computer industry is pretty advanced. And so other people pick up those ideas, and make them into real products. And they, kind of, leave the research projects in the dust if the researchers drop research and start trying to commercialize stuff.

And I can point to 100 examples of that in higher education. So it's probably good that Project Athena has a beginning, a middle, and an end, so that those people don't get stuck trying to do commercial software in an academic environment. It's, kind of, a mismatch.

As far as we're concerned, what we're doing is, there's a lot of labs that are being put in in higher education. And we're winning almost every one of those. And we really go hard after the lab, so that people that can't afford the computers have public access.

And most universities now have fairly elaborate campus-wide networks. It's no longer a new cutting-edge thing. And we plug right into those. So I think it's-- life's moved beyond where it was a few years ago, where those kind of projects were really important. And the knowledge to do that exists fairly widespread. I think a few more and-- yeah?

What's the most important thing that you personally learned at Apple, that you're doing at NeXT?

Good question. I'm not sure I learned this when I was at Apple, but I learned it based on the data when I was at Apple. And that is, I now take a longer-term view on people.

In other words, when I see something not being done right, my first reaction isn't to go fix it. It's to say, we're building a team here. And we're going to do great stuff for the next decade, not just the next year, and so what do I need to do to help so that the person that's screwing up learns versus how do I fix the problem?

And that's painful sometimes. And I still have that first instinct to go fix the problem. But that's taking a longer-term view and people is probably the biggest thing that's changed. And then and I don't know. That's maybe the part that's biological, but--

[LAUGHTER]

Yes?

To pull off on that question, I wanted to ask you about your management style, and specifically, how do you resolve conflict in your organization?

What's our management style? How do we resolve conflict? I've never believed in the theory that, if we're on the same management team, and a decision has to be made, and I decide in a way that you don't like, and I say, come on, buy into the decision. You know, buy into it. Like, we're all in the same team, you don't agree, but buy into it. Let's go make it happen.

Because what happens is, sooner or later, you're paying somebody to do what they think is right, but then you're trying to get them to do what they think isn't right. And sooner or later, it outs. And you end up having that conflict.

So I've always felt that the best way is to get everybody in a room, and talk it through until you agree. Now, that's not everybody in the company, but that's everybody that's really involved in that decision, that needs to execute it.

And so that's how we try to run NeXT. The way we run NeXT is, we have a team at the top we call the Policy Team. There's eight people. Mike is on it. I'm on it. We have six other people on it.

And the key-- we have two things we try to do. One is, we try to differentiate between the really important decisions and the ones that we don't have to make. And the really important ones, we work on it until we all agree, because we're paying people to tell us what to do.

In other words, I don't view that we pay people to do things. That's easy, to find people to do things. What's harder is to find people to tell you what should be done, right? That's what we look for. So we pay people a lot of money, and we expect them to tell us what to do.

And so when that's your attitude, you shouldn't run off and do things if people don't all feel good about them. And the key to making that work is to realize there's not that many things that any one team really has to decide. And we might have 25 really important things we have to decide on a year, not a lot.

So that's how we try to run it. Sometimes it works. And sometimes we're still working on it. I can't think of once-- I can't-- maybe there's once or twice, but I can't even recall a time when I've said, dammit, I'm the CEO. And we're doing it this way, you know? I can recall a time when I've said, we don't see eye-to-eye, and you're off the team.

[LAUGHTER]

You know? I've had to say that once or twice, over a prolonged period of time, when a person has not wanted to go in the same direction we've wanted to go in as a team. It's my job every once in a while to say, hey, you want to go this way? We want to go this way. It's not working. But when people are on the team, then we work it out. Yeah?

You've, kind of, chosen this direction, this niche towards the premium product, versus you've discussed Macintosh going into portability, and a lot of these portability issues. Do you think portability is exclusive? Can you do that and still do some of the market-driven type stuff as well?

I want to come back to your premium product characterization, because our products actually cost close to half of what Sun's do. And the reason they do is, we have the most automated factory in the industry. And we have a great VLSI design group, which designs stuff in a lot less parts. Do you guys-- manufacturing, do you care about that?

Yeah.

Yes.

Yes.

Yeah.

We should talk about that in a minute. The industry is bifurcating right now. And what's happening is, is that the Macs and the PCs as you know them today are all going to be just like this-- hold up your PowerBook.

They're all going to be just like this, only lighter and smaller, before very long. And they're taking the technology we have today, not particularly changing it, and getting it off the desktop in portable forms. And they're giving up a few things for that, but nothing terribly profound.

However, we're getting all sorts of signals from certain parts of the markets that they want things that are the enemy of that, right? Well, what are they? Well, speed-- speed is the enemy of portability, because speed takes power, right?

So the kind of speed that our customers want, it would run for 3 and 1/2 minutes on batteries. And that's useless. The second thing they want is, they want a lot more storage on their disk drives, and a lot more memory, again, the enemy of portability for power and size reasons.

Another thing they want is really high-speed networking, right? The radio LANs on these things, at best, are going to go 19.2 kilobits per second. Our customers want 100 megabits per second and higher. You're not going to do that with a radio LAN anytime in the next five years.

Another thing our customers want is our mix has shifted to, like, 80% color. And they want true color to do photographs. There is no flat panel color display today that will do photographic-quality color. And they want big ones, because once you have multitasking, and you run a bunch of things at once, you need a bigger window into this electronic world. When you're using your computer for two or three hours a day, you don't want to be looking through blinders this big. You want something bigger.

So these are all the enemy of that today. And we're working on smaller products, for sure, but it's really tough to get both. And we're optimizing for the power, because we see a giant hole there for running these mission-critical custom apps. And what people are doing needs more power, so that's what we're optimizing for.

Let's talk about manufacturing. How many of you have a manufacturing background? Oh, that's great. I love manufacturing. And what kinds of things? What kinds of companies? You have--

Pharmaceuticals.

Pharmaceuticals? Any auto people here?

Yeah.

Him.

Auto? Electronics? Which

TI.

TI, uh-huh.

Motorola.

Motorola?

Is it true you're sending your manufacturing overseas, or just-- I heard--

Yeah, I heard that rumor, too. No, it couldn't be further from the truth. We love manufacturing at NeXT. And when I was at Apple, I had the good fortune to lead the effort to build a Mac factory. And we designed, and built, and operated that factory. And it was a real breakthrough. It was the best factory in the industry until we built the at NeXT. And we made a lot of--

[LAUGHTER]

We made a lot of mistakes, though. As an example, I remember walking through it. You know, one of the things you learn when you start building factories is that warehouses are really bad, right?

Warehouses are bad, because you tend to put things in them. And inventory is really bad. Inventory is really bad, because if it's defective, you don't find out about it for a while. And you don't close the quality feedback loop with the vendor, and correct the problem, until they've made a zillion of them. What you want to do is find the problem the first one that comes in the door, and stop them from making more until you fix the problem.

So warehouses also cost money, because you put all this stuff in them. And the stuff-- you have to go borrow money from the bank, or use money that could be used in a more productive purpose, so warehouses are bad.

And you want to go to JIT. I'm sure you've studied this all, and studied examples. I was walking through the Mac factory one day, and the two biggest pieces of automation we put in were a giant small-part storage and retrieval system. It was the totes that ran around.

And the second one was this giant burn in system at the end. And a few tens of millions of dollars worth of equipment. And I realized, unfortunately too late, that both of them are warehouses. They're just high-tech warehouses.

And so when we looked at NeXT, we said no warehouses of any kind. We have a true JIT factory. Stuff comes in, and is delivered right to the point of use on the factory floor. There is no warehouse. Deliveries are made daily, sometimes more frequently than that. There is no outgoing warehouse. Everything is visible.

And the reason that we were able to do a lot of what we've done = because we looked at-- well, I'll give you an example. When we were learning about manufacturing at Mac, we hired a Stanford Business School Professor at the time named Steven Wheelwright, who Harvard has since stolen away, I think.

And he did a neat thing. He drew on the board a little chart, first time I met him. He said, you can view all companies from a manufacturing perspective this way. You can say there's five stages-- one, two, three, four, five. They all have these things.

And stage one is companies that view manufacturing as a necessary evil. They wish they didn't have to do it, but damn it, they do. And all the way up through stage five, which is companies that view manufacturing as a competitive opportunity for competitive advantage, right? We can get better time to market, and get new products out faster. We get lower costs. We get higher quality.

And in general, you know, you can, sort of, put the American flag here, and put the Japanese flag here.

[LAUGHTER]

[APPLAUSE]

And that's changing, however. That's changing. And it's changing, because people like you are going into manufacturing. Companies are starting to realize that we were great at this one time. And then we took it for granted. And people are starting to pay good salaries now, and get good people.

And so we want to be one of these. And we try very hard. By the way, just going back to software for a minute, I often apply this scale to computer companies, and how they look at software. See, I think most computer companies are stage one. They wish software had never been invented.

I put Compaq in that category. And IBM is maybe stage two, and things like that. And I think there's only-- you know, Suns maybe, sort of, in the middle, maybe here. I think there's only three companies here, and that's us, Apple, and Microsoft, in stage five. We start everything with the software and work back.

But anyway going back to manufacturing, we started looking at the factory as a software problem. And the first people we hired in the factory were some software engineers. We convinced them to move from R&D into software, which was not easy.

We had to give them bonuses. We had to cajole them. We had to promise them they could come back if they hated it. And they went over there. And we said, this is really just a software problem with interesting I/O devices called robots. That's all it is. And so we started building the software first.

And our first robots that we got, we specced them out. And we bought them completely turnkey, with the robot arms on them, and all the electronics, and the software to control them. And we specced it out, but we didn't write it.

And they didn't-- they worked OK. Some of them are still in use, but they weren't great. And being software folks, we weren't real happy. They weren't elegant. We couldn't do what we wanted with the robots. We couldn't tie-in a quality information system to them, and all this other stuff we wanted.

So the second generation, we specced out the hardware, and had somebody build the hardware for us, but we wrote all the software on our own computers. We're object-oriented, so we started writing robot objects, quality objects, you know, all sorts of objects to control this factory.

And we found in our computer was great for it. And so our whole factory now runs on this object-oriented factory and quality system. The last generation of-- our latest generation of robots, which we've deployed this year, we actually built the hardware.

I've been to Japan maybe-- oh, a lot of times-- maybe 30, 40 times. And I love to have factories over there. They always amaze me, because they built everything themselves. They weren't afraid of anything. They needed a robot. They tried to buy one. But if they couldn't, they'd actually engineer it and build it.

And you'd think this was really expensive, but we found out it's pretty cheap. It's actually cheaper than buying them. And so we've actually now designed our-- and specced out our own robots. We don't mill the metal or anything. We get that all made. We put them all together. And we do the software top-to-bottom.

And we have now some extraordinarily advanced robots in the factory. And our computers are built, start to finish, on the key components, completely untouched by human hands. So we're pretty convinced we're the low-cost producer. We do it in Fremont, California, right under our nose. And we export them to Japan, and all sorts of other places.

And Canon is our partner in Japan. And they do very, very thorough quality audits. And we're now at the point where we're directly shipped to stock with them. And they say we're a very high-quality supplier.

How do your lines tie in with your research-- your development team? Because I had heard that they could actually change the line from their own computers [INAUDIBLE]

Yeah, they can. Well, we don't give everybody permission to do that, but

[LAUGHTER]

--yes, they can. Here's how it works. One of the things we do is we actually-- when we want to build an engineering prototype-- see what happens in most-- one of the key things that manufacturing can contribute to competitive advantage is time to market. Why is that?

Because the way most things work is you design your product here. And after you're done, you throw it over the wall. And you design your manufacturing process here, sorting out a bunch of things that maybe weren't done right here, fixing them, changing them, and then completing the process design.

What you want to do is do this, and ship it right here while your competitors are still here. And that's what we've been able to do in many cases. What we do is, we suck data out of our CAD systems in engineering. We zing them around over the local networks over a T1 to [INAUDIBLE] factory is about 15 minutes away.

And in our own computers, we compute all of the robot placement programs' fully-optimized path. We compute all the vision system programs. We check it against the bill of materials in the IS system. And we download it to the robots. And we're ready to build a board, lot size of one, in-between two production CPU boards on the line, full surface mount with all of our automation technology.

Now, the key is that manufacturing did that so well for engineering, that we haven't built a prototype in engineering for two years. We haven't built a wire wrap, or any other kind of prototype in engineering for two years. Everything has been built in the factory.

Now, what does that mean? What that means is, manufacturing gets involved from day one. Because the-- the engineering guys call up manufacturing go, hey, we want to build a prototype. We're going to need these special parts in that thing. Take a look at this. Tell us what you think. We'd like to do it tomorrow. Let us know if that's OK, blah, blah, blah. They get involved from day one.

And what it also means is-- so we get this parallelism. Secondly, a lot of times, when you build prototypes, it's not quite the same technology as you're going to use in production. And so all the accumulated knowledge you get from building your prototypes, you throw away when you change technology to go into production. And you start over in that accumulation process.

Because we don't change technology, we don't throw anything away. We don't waste time. And it's led to one of the healthiest relationships between an engineering and manufacturing group I've ever seen in my life.

They're all working off the same databases. They're all working on the same processes. They're all working in a very disciplined process environment, to where, when any processes are change, they all get together and review the proposals, and all buy into it.

And it's not that hard. The key to it all, though, was we didn't go out and hire a bunch manufacturing people. We went out and hired engineers. And we convinced them that we were going to be different. We were going to pay them exactly the same as-- as a matter of fact, we paid them a little more at the beginning.

But we pay him exactly the same as R&D, no different. There's migration, both directions, not just from manufacturing into R&D, but both directions. And they're not second-class citizens. They have the same offices. They have the same test equipment. They have the same computers on their desks.

And it took us a while to convince them that we were really serious. For about the first few years, we had more PhDs in manufacturing than we did design engineering, until design engineering stole a few of them away.

So it's really paid off for us, and I think it's one of our real opportunities for competitive advantage. Yeah, I think one or two more, and we've got to run. And probably, you do too. Yes, up there, [INAUDIBLE]?

So you have no warehouses, is that true? You are doing true JIT? [INAUDIBLE]

Yeah.

How are you getting your products or your raw materials?

How are we getting our raw materials?

Yeah.

You mean, like, what truck line brings them in? Or what do you mean?

[LAUGHTER]

Is it air? Is it truck? I mean, how are you getting everything in that quickly?

See the key thing is, that's not our problem. That's our suppliers' problem. So we agree with our supplier when the stuff is going to arrive on our factory floor. And if they can-- if they're together enough to ship it by truck, that's fine. If they have to ship it by air, that's too bad. If they want to have a warehouse next to ours, because they're not good enough, well, then they have to do that.

Now, we're not giant, so we can't go command people to do things. But what's happened is, is we have a fairly narrow supply base. We don't have three billion suppliers. And they see tremendous advantages in working with us.

We're pushing our quality information systems back to them. As an example, Motorola is one of our key suppliers. Almost every key supplier has NeXT computers. And we send them statistical quality information, sometimes daily-- daily-- off our automated quality information systems on their parts.

And those kinds of things are extremely valuable to them. So while we're not Goliath, we're a very valuable David to work with. And so they really bend over backwards to work with us.

And we try to push the problems where they belong. If it's our problems, we take full responsibility for them. We own our process. But they-- it's their job to get us a zero defect material on-time, per agreements. And our philosophy is, our money doesn't break after we give it to them, so their parts shouldn't break after they give them to us.

[LAUGHTER]

[APPLAUSE]

Yeah?

Do you see any possibility of bringing NeXT over to [INAUDIBLE] portable Macs? And do you see the future of Macs [INAUDIBLE] dying out, or what [INAUDIBLE]?

You know, I think Macs are going to continue to grow. It depends on what Apple does. I think certain segments of the Mac market are going to continue to grow. I think certain segments of the Mac market are not going to be targeted for future growth.

And I think Macs are good computers to do certain things. But to do the things that people I think are going to want to do in a few years, I don't think you can do them on a Mac. I don't think you can do them on DOS and Windows.

And so I think there's a need for some new technology. Just like, you know, there a lot of things that you can do on DOS when the Mac came out, but there were some new things you just couldn't. And it would take a long time. And I think it's the same way with the Mac. Hey, we have to head off into the sunset back to California, but thanks for a chance to be with you all for some time.

[APPLAUSE]