MITOCW | ocw_esd_57_2007-10-11.mp4

IRVING WLADAWSKY-BERGER: Last week, we started talking about execution in the marketplace, and we talked about e-business and how a major part of what we did in IBM to help formulate a strategy that we could execute was to leverage our base, which included base of talent, the base of clients, existing products with the new stuff.

And that the way we framed it, and it was a very good framing, I think. Well, it turned out to be pretty successful in the marketplace, is for IBM, the internet, World Wide Web, and e-business meant leverage the internet to make what you have that much more valuable. And that's our framing. Other people would have framed it different, saying, what you have goes away, get rid of it. That's a fine way to frame it. You have to convince your customers.

And we talked about some of the key aspects of organizing this strategy, the market strategy, which is balance between proprietary and open, in-house versus partnerships. What are going to be your key offerings? How are you going to measure your progress? And those are very key organizational things.

But we also talked about a very important subject, that when you're talking about a disruptive technology, like the internet, with a very big potential market-- and remember, we talked about the goddamn internet is just this incredible system.

And as you know, if somebody said, well, where in this stack does the internet play? What makes it so disruptive is that it plays up and down. The more you go across, how do you focus your efforts?

If you have something that complicated, if you just said, well, I want to go take on the world and I want to go and transform applications, no, no. Take it easy. We may get there. But the first question is, what are you going to do first? How do you get to first base? You know what I'm talking about. Let's get there one step at a time.

And we will talk today about customer-based experimentation. And first, I'll talk about the work of Tom Key. And then we'll talk about Eric von Hippel's work. And Eric was really nice in saying he'll join us at 2:00 o'clock, which is wonderful, because he will then join us in the discussions.

I can't say enough about experimenting in the marketplace as a way to figure out what to do. If you're just doing pure technology invention, clearly you do that in the lab, and clearly you experiment in the lab. And if you're doing products, you would clearly prototype-build, et cetera.

But innovation, especially now, more and more, and what we're talking about in this class, is how do you take stuff from the lab or from somebody's lab-- don't have to be your lab. Could be MIT's lab, Stanford labs, anybody's labs-- take it into the marketplace?

Now, the marketplace is a wonderful concept, but it's a cosmic concept. So you need to start narrowing down, what the hell do you mean by the marketplace? In other words, you have to tell me where is first base, so that we know where you're heading. And a critical part of finding first base is marketplace experimentation. Working very, very closely with clients.

Now, in the past, marketplace experimentation was just too expensive. There was very difficult to do. First, it was very difficult to build anything in the lab, then to build a prototype. Remember, most of the channels to market, most of the dealings with clients-- not most, all of them were people oriented. There was no technology to help us do that. So it was just very expensive. So you had to be very careful.

And often, the market experimentation was another way of you announce your product and you hope and pray the marketplace likes it, or your service. But this is changing. Companies now have an opportunity to innovate and experiment in a whole new way.

What are some of the things companies can do to experiment? Well, here is a long list. First, there is the internet itself, which as you know, is the world's biggest platform for innovation, experimentation, and collaboration. Then is the fact that IT systems are so inexpensive.

I don't know if you know, but the amount of money that, let's say, brand-new ventures need to get going now is significantly less than during the dot com days. And a big part of that is that during the dot com days, in order to get computers, they needed to buy all this stuff, and that stuff was much, much more expensive.

Today, computers, much to the chagrin of companies like IBM, are dirt cheap. And especially to get going-- and you can get software for practically nothing if you use LEMP and LAMP, and other open source stacks. So the cost of getting going is much less expensive.

Even supercomputing, which before you had-- it was very expensive to access. You can do that much better. And on and on and on. So there is a whole set of new tools that very much enable you to experiment.

And Tom Key makes the point that companies need to take that into account the opportunity for experimentation in their basic structure. That is, it's not like I go with the way I've been doing it for the last 50 years, and I'll bolt experimentation at the end. That's not the right way to do that.

He makes the point that you want, from the beginning, to organize your efforts in terms of extensive experimentation. What is an example? Well, if you have a huge team that needs to do something for production, which maybe is what you need, maybe you want to have a small black belt team that does a rapid prototype that you can even start making available as an alpha or a testbed to start testing out the ideas, and then incrementally refine what you build so that you can start experimenting very quickly.

I have to tell you, in the old days, people could bring, say at IBM and other companies, product plans that may have taken, I don't know, three to five years. Today, a plan that three to five years-- it's not that it would be a joke.

And if you're building a new semiconductor fab, that's a reasonable plan because you're building a semiconductor fab. But let me tell you, if you're building a software product or an application or a new service, three years, it's ridiculous. The market will have changed several times.

So you have to think what you plan to do. I advise people, tell me what you can get done in six months. Well, I need-- no, no. I know you need more time, but what can you get done in six months? And then do that quickly. And then six months after that, and six months after that, you keep iterating. So it really changes the whole way you look at what you're building.

It doesn't work in every case. I don't mean this is universal, but I'd be very worried if somebody says, Irving, it's three years or nothing. I'd be very, very worried. Please.

AUDIENCE: What you're just describing, it seems pretty obvious in the IT and software industries. How universal is it?

IRVING WLADAWSKY-BERGER: Well, let me ask you, who have experience in other industries, drug research in pharmaceutical? There's a lot of work about testing new pharmaceuticals much, much faster with all kinds of new technologies. Do you all want to talk about your experience in your industries?

AUDIENCE: In the drug industry, hand skills are to get out the drug are very-- the product plan would be something at least 10 years ahead, but the emphasis today is on failing really fast. And I don't know if this is connected with how fast we learn. But it seems like if you iterate and you iteration fail and takes three years, you tend to forget as an organization, what drove your iteration. Whereas if it's about six months or three months, you tend to keep that--

IRVING WLADAWSKY-BERGER: That's my feeling. And are pharmaceuticals making progress in doing that?

AUDIENCE: They're trying. However, the problem is every iteration costs a huge amount of money. So the question is between--

IRVING WLADAWSKY-BERGER: I understand.

AUDIENCE: --balancing money and your knowledge.

IRVING WLADAWSKY-BERGER: I understand. How about other industries? I mean, in retailing. I heard this talk in June, the CEO of Metro. It's a big, big store. And we'll hear from Chris McCormick in a couple of weeks. We can ask him that. Seventy percent of new products on the shelf fail.

By fail, I mean, customers don't buy them. It doesn't mean-- the product may be wonderful products. It's just if you define success as customers buy it as opposed to they are pretty and customers should buy it, then 70% fail.

And this may get back to one of the questions that he mentions. Oops. If something is going to fail, if it's inevitably going to fail, the earlier you find out, the less-- in fact, if something fails quickly, it's a learning experience.

If something fails after you've invested tens of millions of dollars and years, it's a goddamn problem. So in some ways, what we're talking about is, is there a way to do this so that whatever doesn't work you find out quickly? Because maybe what you're then doing is maybe weeding out a lot of the efforts, and you can concentrate on those parts that are most promising.

Maybe that is what we're doing. But I know there are a whole new drug sequencing technologies that even in wet labs allow testing of lots of stuff. Of course, genomics sequencing is coming way down. Simulations are enabling to view designs at a much faster way than ever before supercomputing simulation and so on.

But my feeling, and I really want to hear about your industries, is this may be one of the biggest innovation areas for every business. How do you reframe the way you do things to take this approach, or start with something and then start refining and incrementing with the biggest opportunities? And it's not clear what to do, but do that.

Now, this gets back to some of the things we talked about last week. Somebody said, oh, Irving, you don't I have to build this whole infrastructure. I said, well, what's wrong with the internet and open source? Well, I don't like Linus. Linus is a fine guy.

And so the more you leverage what exists-- an open source has the nice characteristic if you can use it as a platform, that the price is right when you are starting out. It may not be great for scaling later. You may need to do things for scaling. But to get going--

And if you use as much existing components as possible, obviously you don't want to steal anybody's IP, you just start way ahead of the game. You don't have to start from the bottom. You start from quite a bit.

If you partner with others that have a base you can build on, a platform that you can leverage and then build on top of that, you can just go to market that much faster. But it's an interesting way of thinking. I know at IBM, I think I have mentioned that. I would say 10 years ago, people would think, well, you don't think I'm good enough to build the whole thing, that I have to go build it on Java, which comes from Sun, or I have to go acquire Lotus.

Well, you're good, you're wonderful. But it will take you three goddamn years, maybe five. And we have zero promise that after these 3 to 5 years, you'll be where they are today. Zero.

So from a business point of view, that sounds dumb. So the more you'd say, OK, what's a base that lets me-- it's a base that's not-- that is an enabler for your innovation that lets you get going? The faster you find an existing base, the faster you can get going.

And I think that attitude permeates businesses these days. That's why initial investments are so much lower than they used to be, because they expect everybody to use a base. Any comments on this point from all your experiences?

Now, the third point that Tom Key makes is to-- it's the same point, essentially, we just made before. By low fidelity, he means the early designs don't have to be the ultimate designs. They can be course designs that will show later that you will keep incrementing later.

So it really keeps-- it's another way of saying pick approaches to design that enables you to incrementally improve them as you get information and feedback from the marketplace. That's what he's saying. And that's another way of saying that.

And then finally, and again, we just discussed that. Be very clear about what your unique innovation is. What is really new in what you're trying to do? That's what you should focus your energy. That's what you should take the risks.

Now, if probabilities of failures multiply, and p1 times p2 times p3 quickly get very small. So part combine new and traditional is if you build something new on top of existing components that are well understood, where the risk is minimal, you're just increasing your overall probability of success.

And later on, once the whole thing is better, one of the first things you might do for release 2 is improve this component and get a better one. But now you have a stable product and you know where it fits.

One way that I have thought about it often is that when you have a project-- think of a project and think of-- I have this mental image, lots of moving parts in the project. Well, if you have too many moving parts, the damn thing can never come together. And if the thing doesn't come together, unfortunately, it falls apart.

So you want a kind of core around which the moving part can settle down so that you are incrementally growing, and then come back and start replacing the core. And that's a way of reducing risk. Please.

AUDIENCE: It seems like you're anticipating early information and fail early. You're saying, put the 80% solution out there. And you have to be careful with that, I think, because if you put out there an 80% solution but everybody doesn't know it's the 80% solution, or if it's really the 70% solution and everybody gets pissed off at you, and then even if you improve it, you have lost the credibility.

IRVING WLADAWSKY-BERGER: I agree, totally. You have to be very careful.

AUDIENCE: There's a line there--

IRVING WLADAWSKY-BERGER: Totally. No, no. But remember, if we knew where the line is exactly, business wouldn't be business. No, no. I think a lot of the innovation is picking the line. And I don't know-- I have a lot of emphasis on design in everything that I keep talking about, and the marketplace.

And a lot of the reason-- and by the way, I mean design and the marketplace for the engineers, not for the MBAs that are doing whatever MBAs do. MBAs need to do-- we need to count money. We need to do a lot of stuff.

But the kind of guidance about how to do a design that is suitable for the market you're after, where the 80% you do is correct. That has to be intertwined with your overall design. And if you are the design manager, chief architect, and if you have very good intuition about it, you can do a much better job than if you have to listen to what somebody else who's been out there tells you. But it's a huge part of the innovation in my mind.

AUDIENCE: But also failing does not mean having to put it out to the public. Like a drug, I wouldn't test it on a human being.

IRVING WLADAWSKY-BERGER: Yeah, I understand.

AUDIENCE: Or cars.

IRVING WLADAWSKY-BERGER: Yeah. Also, it may be simulation. Remember, in a lot of areas, the way you do early experimentation is massive simulation.

AUDIENCE: You said that the drugs--

IRVING WLADAWSKY-BERGER: No, in drugs. But in cars, you might be able to do that. You might be able to design the car, start testing its aerodynamic qualities, its manufacture. And the amount we can simulate keeps going out and out and out.

So out there is not necessarily customers in the usual sense. But we'll talk in the next section, often it is. And of course, there the key is to select the right customers. Please.

AUDIENCE: Failure, by nature, it is defined by market. I mean, that's the way you define. So how do you not go to the market and discover failure?

IRVING WLADAWSKY-BERGER: No, no. To discover market failure, you have to go to the market. To discover that this car just has bad crash properties, that you can do quickly. Or even, let's say, to show people-- and that's why Detroit does concept cars. And actually, not just Detroit. Everybody does concept cars.

And I think a lot of the reason for the concept cars is find out if people go bananas when they see the car, or if they say, this is the ugliest car I've ever seen and they don't want anything to do with you.

And again, my view is through extensive use of simulations modeling, you can probably get a concept car to a show much faster than in the old days where you had to rely on modelers. Isn't that correct? There is a tremendous amount more you can do today. So there's a whole set of techniques, but all of them are getting out there as quickly as possible.

AUDIENCE: What kind of experiments can you do in the market to produce market changes? That's really frontier, actually.

IRVING WLADAWSKY-BERGER: It is. And let me wait 'till the next section, because we'll talk about that very much in the whole next section. I agree totally, that is the frontier. And we'll talk a lot about it. Yes, please.

AUDIENCE: I mean, in software, which is where I come from, lots of prototype mockups, you run it by users and you just keep doing that very, very quickly in the process and then throw it out for you. At the end, you have made all the mistakes very quickly and validated with at least some of the public from the consumer that it is not working. Even if that's a feasible model, you just throw it out.

IRVING WLADAWSKY-BERGER: Well, that's why alphas and betas are so important. That's why a-- I think John Patrick mentioned we developed this notion of alpha works because we had all these ideas in our research labs that had no channel to be experimented within the market.

So we said, well, screw it. Make it downloadable. And the first time we told that to the lawyers, they had apoplexy. And we said, no, no constructors and IP something. Will you sign here and say you will not use this for a commercial purpose, only for experimenting? And we did that. So that's a vehicle to get it out there.

And software, there is a lot of experimentation of that sold. I mean, any of you that deal with Yahoo, Google and so on, you keep getting gadget-- I mean, there are names for-- what are the names? Gadgets and widgets. Am I correct?

AUDIENCE: Google gadgets.

IRVING WLADAWSKY-BERGER: Google gadgets and Yahoo has widgets, right? And I don't know what Microsoft calls whatever they have. Do we have an IBM widgets and gadgets?

AUDIENCE: [INAUDIBLE] sometimes.

IRVING WLADAWSKY-BERGER: So there are-- in software, there are. Now somebody says, oh, can I run the Federal Reserve money transfer? We say, please don't. Or can I run a surgical operating room with that? No, we don't recommend that.

So you have to be very careful about who you pick. Otherwise, your whole brand goes up in smoke if you pick the wrong sections. But you pick the right sections, and in the next section, we'll talk a lot about it, it's good.

AUDIENCE: It's interesting you brought up Google. They use beta. They put beta on everything. In fact, I think Google Maps is still beta.

IRVING WLADAWSKY-BERGER: Is it still beta?

AUDIENCE: Google Mail is still is. Everybody's using Gmail.

IRVING WLADAWSKY-BERGER: Gmail is beta?

AUDIENCE: Yeah, everything's beta.

AUDIENCE: Yeah, they call it all beta. It's a marketing spiel. So if something goes wrong, like you don't get too angry at beta, right. So they get around to on all of that.

IRVING WLADAWSKY-BERGER: But remember-- let me ask you a question. How much do you pay for Gmail? No, no.

AUDIENCE: But advertisers are paying.

IRVING WLADAWSKY-BERGER: No, no, I know that. I know that. No, no. But consumers--

AUDIENCE: Yahoo has the same business model. Yet when Yahoo Mail goes down, they get all kinds of irate users. I mean, it's a marketing strategy.

AUDIENCE: Yahoo Mail was made up for two years before anything. They did the same thing.

AUDIENCE: But Gmail has been around for four years or so.

AUDIENCE: The key to Gmail is that their recovery process is-- Google can-- they put code out every week. But if there's an issue, that rollback is within an hour. That's the key to what they are. So they're able to do production.

IRVING WLADAWSKY-BERGER: So that's equivalent of very quick failure and recovery, because you are expecting that anything goes wrong, you'll immediately roll it out.

AUDIENCE: Or is that rapid experimentation?

IRVING WLADAWSKY-BERGER: Well, it's both. I mean, again, remember this gets back to why I want-- I think it's so critical that the engineers be very involved in marketing, because we're talking marketing. But this is the kind of marketing that can have a huge impact on your architecture.

And if you don't have the right-- you cannot do the rapid recovery, somebody says, well, I know you build this architecture where everything is monolithic. There are no components. I want you to do rapid recovery. I don't think you can do that, right? It has to be built into the basic architecture, which is why these feedbacks are so important. You had a point?

AUDIENCE: Yeah. I just wanted to make [INAUDIBLE] was in perpetual beta so you never come out of it. You take it out to the marketplace like this and then figure out if-- for example, Gmail they put out a new thing almost every day.

IRVING WLADAWSKY-BERGER: No, that's right.

AUDIENCE: Are they OK with it? And keep on--

IRVING WLADAWSKY-BERGER: Right, but remember, it depends on the product. I mean, it's very important to know that a hospital management system cannot be in perpetual beta. So you need to be careful about what products you need to use. Let me call it more classic development cycles. And even the classics, you can innovate, and what products are in the perpetual beta category. And in fact, I think that one of the critical innovations when doing any product is to look at the overall-- and when I say "product," I mean "project" because it can be services.

And almost analyze task by task and do a kind of triage. Is this part of the mission critical? Then it goes into this bucket. But if there are tasks that are not in that category, then you can, in your design, put them in a different portion and allow experimentation here and not in the ones that have to be mission-critical. And that's a big change in design to do that kind of careful analysis of the tasks. Please.

AUDIENCE: [INAUDIBLE] two comments. One, Flickr was in gamma for [INAUDIBLE] and--

IRVING WLADAWSKY-BERGER: Was in gamma?

AUDIENCE: Gamma. Flickr gamma [INAUDIBLE]. I would just add that Oracle who does the same sort of thing as Google. They're offering SaaS-type delivery now. And they're adopting the same Google model for mission-critical apps. They're putting out incremental small patches every single day with a back [INAUDIBLE]. So it's not just to please or avoid criticism from the customer by putting out the beta [INAUDIBLE] make any difference.

IRVING WLADAWSKY-BERGER: Very good. Oops. So now, let's switch over to democratizing innovation, which is available for nothing from Eric's website, which is a question you can ask him how come everybody's selling their book? He has a hard copy of it, which I would encourage at least some of you to buy. It's a very nice book, but if you decide you don't believe in buying what you can get for nothing--

[LAUGHTER]

--you can download it off his site. And he makes a very big point about the changing nature of innovation and the distinction between manufacturing-based innovation or classic innovation that takes place by a manufacturer and user-centric innovation, which I think it's a little bit easier to talk about in the next slide.

In manufacturing innovation, what happens is if I'm the developer, I get all the information, then I go on, develop the product by myself, build it, and so on, and then I push it into the marketplace. That's the classic approach to innovation. And that's been used forever. In user-based innovation, the people using the product themselves are participating in the innovation, sometimes with the support of the manufacturer, sometimes without the support of the manufacturer, whether it's hacking or doing whatever they do.

And what Eric has observed is the growth of user-based innovation, how important that has taken place in the old days. In this case, the users are passive-- they buy whatever you produce. In this case, the users are very, very active. And where user-based innovation is most important is in the very early stages of a potential product, where you don't even. It's almost pre-alpha.

You don't know what it is that you are going to be doing. And so you very much want leading-edge users that have the interest to work with you to participate and help you. And once things become clearer and you are in the marketplace looking to scale up production, scale up support, and deal with users who don't care to innovate with you, then the more classic manufacturing principles take over.

And this is extremely important, this user-based innovation, especially when you're dealing with projects of the kind of complexity that internet-based projects tend to have. Let me give you an example from my experience. Before I led IBM's internet project, I was very involved in supercomputing in IBM. And in particular, we started a whole line of parallel supercomputing in the early '90s.

And at that point, that was happening at the same time as the massive bipolar to CMOS transition we've talked about. A similar transition was happening in supercomputing. And before that, Cray Research was the leader in supercomputing. And they had really exotic technologies. But their machines were incredibly expensive. And parallel architectures based on initially risk technology started showing up and gave tremendous power at a fraction of the cost.

We honestly didn't know what the hell we should be building. And what we did was pick a couple of key customers to be our launch customers and do something just for these two customers as quickly as possible. Now, the two customers we picked were the Argonne National Laboratory, which got DOE funding to participate here, and the Cornell Supercomputing Center, which is an NSF center.

Well, when you're dealing with universities and research labs, it's been my experience that they consider it an insult if you don't trust them enough to have them as lead users. In fact, before the new IBM era, post-near-death experience, IBM didn't know how to deal with them because products were products. And we say, no, here is the product. And they would say, well, but it's finished.

Well, of course it's finished. We only do finished product. But we don't like finished products. That only means you don't trust us. You don't think we should participate. And the answer was probably, yeah, we don't really trust you because you're a pain in the ass. And you're so different from our classic commercial customers. So maybe we won't do business with you. And by the way, I'd pay the huge prices of being removed from tremendous innovation, which IBM did, and others would do that.

But not only did the Argonne Lab and the Cornell Supercomputing Center were our clients, but in fact, the reason they got their funding from DOE and the NSF, respectively, is because they went to make the case that they would work with IBM to influence IBM's design. So that was a feature of what they were doing, because that's what a National Lab should do-- have an impact on the private sector. And that's what an NSF supercomputing center would do.

And the first machines shipped in April. And essentially. I mean, they were really not very good. By that, I mean, they were incomplete. And you had to do a lot of hand-holding. People from the lab had to go there. But you're dealing with national labs. Hey, these people, what's a little assembly language programming or doing hand-coding parallel architecture? In fact, they almost became an integral part of our development process in doing this stuff.

And the work with them allowed us to then release the first version of the product in a few months later. I think it was in October of '93. We shipped the first machines to them April '93. Then we shipped the first version of the machine in October. And at that point, we had a few more customers, including some commercial customers. But in supercomputing, commercial customers means petroleum companies and engineering companies.

And then the following year, in '94, we really announced the product. Became known as the SP2. And by the product, I mean any kind of classic cycle would say, you don't announce until you have this. But by the time we announced the SP2, we had been in the marketplace for a year. And it's almost like our people in the lab And even our people in the field had been in spring training in, how do you work with-- remember, there's a whole process to how do you support customers, hold their hands, how many people you need?

And if you started practicing that once you ship the product, you're in deep trouble, because the customers will want you to do. But by picking the right leading-edge customers, all of a sudden, it just plays into your marketing. It just totally plays into your marketing. I've talked about an example for a new product and a new pretty research-oriented product.

Now with commercial products, whether it's mainframes or other things, the equivalent is to pick selected clients with whom you have a very, very close relationship. And that you typically pick five of them, would you say, or so, who become your early clients, who've been dying for the features that this release is taking to market, and with whom you have such a close relationship that even if they are really upset at you, they won't go write blogs about it.

But they may go back to the original release, but they would not embarrass you. And so you pick them very carefully. And you work very carefully with you. A big part of what you do here is you are supporting those early adopters directly from the lab. So I mean, here, you can do that. Here, you need the whole field support mechanism and everybody else. But in fact, I think what usually determines how many of these you can do is how many can you afford to support out of the lab, and people flying in and people doing all kinds of things. And then you move in this area.

Now, I would expect in the pharmaceutical companies-- I mean, let's take somebody who's doing an experimental drug to cure a certain kind of leukemia. And there are patients that they've tried everything else. And can you get permission from the-- not the FAA, the FDA to--

AUDIENCE: I guess you have accelerated clinical trials. [INAUDIBLE].

IRVING WLADAWSKY-BERGER: So you might be able to get special permission.

AUDIENCE: [INAUDIBLE] for phase 1. It's safe. And you might be able to do that. You might be able to combine that phase 1 safety on these patients with these patients, so that [INAUDIBLE] they are faster than the market and faster than the [INAUDIBLE] result. And that's what actually we ended up doing with a current run [INAUDIBLE] clinical trials. We tested on a bunch of patients for whom the traditional therapies did not work.

IRVING WLADAWSKY-BERGER: Yeah, so that's an example where these people are superb leading-edge users. And they're the people who take care of them should be big innovators, right? I know people have said-- there's this famous hospital in Houston, the Anderson Hospital. I think it's the Anderson Hospital. If you go to Anderson, that's where they do the most advanced research. But it's not good to land in Anderson.

[LAUGHTER]

Because, I mean, it's not a joke, but It means all the other things have not worked. So you are now landing-- I mean, for what it does, it may be one of the world's-- the first heart transplant by DeBakey was done at Anderson in Houston. But you don't want to go there. You want to stay away from that. So this is a very interesting paradigm and approach. And by the way, as we are talking, keep assembling questions to discuss with Professor von Hippel when he gets here.

Now, when IBM in the '70s and '80s decided, for a variety of reasons, to concentrate on just more pure commercial markets and commercial customers and avoid universities, avoid research labs, because we didn't have those products, we could say, well, there was a segment you had. But we knew it even then, but in retrospect, it's clear. What we were doing is cutting ourselves off totally from this incredible source of innovation.

So it's well accepted that the supercomputing community is working on the kinds of products that, a few years later, will gravitate toward the general marketplace. And if anybody wants me to give you exhibit A, how about the internet and world wide web, which all came from the supercomputing community? They were the first users of both of them, let alone parallel architectures, the browser, and on and on and on and on.

So these lead users are incredibly important. Because no matter how good your labs are, your own labs are different than the users'. They tend to be more developing technologies, developing products. And your people in the field are usually, one, not users, and two, they may not be as sharp as the users. They may be brilliantly sharp people. It's just that these people know their stuff so much better.

So cutting yourself off from dealing with these users is a sin. I mean, not a sin-- it's bad business practice, because you're cutting yourself off from one of the biggest sources of marketplace. Let me give you another example-- marketplace innovation. A few years ago, I had just become the chairman of IBM's Academy of Technology. And some people came to see me to give me the results of a major study that the Academy had conducted.

And it was the impact of game players and related technologies. And they said IBM has to be involved in that. And I had no idea what they were talking about. But this is the Academy, so you listen. And essentially, the point was that in the same way that supercomputing is foreshadowing where fields are going, a lot of what's going on in game players, a lot of what's going on in massively multiplayer online games, *World of Warcraft,* and then role-playing games like *Second Life,* are leading-edge indicators of where things are going to go. So I took that very, very seriously. And then we charter a few additional studies, which eventually led to IBM becoming very active in virtual worlds, and things like that.

Another manifestation of this area is what you see happening, let's say, in social networks, where most of the time in social networks, Myspace and Facebook and so on, as you know, I think Microsoft is going to buy 5% or 10% of Facebook for $1.5 billion. But Facebook was done by a couple of kids from Harvard. I don't think it's clear which couple of kids. I think there are some lawsuits going on about that. And I don't know where Myspace came from. And Craigslist was done by-- I forget his first name, Craig. Craig. Craig is his first name?

AUDIENCE: Newmark.

IRVING WLADAWSKY-BERGER: Huh?

AUDIENCE: Craig Newmark.

IRVING WLADAWSKY-BERGER: Craig Newmark? And this is a very good example of user innovation, all these areas that these people, whether it's Facebook in the Harvard example or the other ones, they were just done for themselves. And they started to do more and more and more. And then they found there was a market here. And eventually, the commercial world discovered there was a market there.

But what's very important here, you are a business, and you're not watching what's going on in those people who owe you. That's all I can say. I mean, something can happen, and you'll be in deep trouble. Now we've talked about a number of things here that lead users, by definition of lead users, are at the edge of this market. So in '93, 92/93, our lead users were the people most interested in parallel architectures.

Our lead users now in supercomputing are interested in petaflop supercomputers. Petaflops, just enormous machines in Los Alamos National Lab, University of Illinois NSF Center, Argonne National Lab continues to be there. So these are the people that are just pushing the envelopes at every single level. They just want to keep the latest, and they will only partner with you and be your lead users, not your competitors, if you're willing to help build what they are looking for. So this is very, very important. By the way, you may decide you can't afford it. You may decide it's diverting your energy. So these are things you have to do very, very carefully.

Another point that Eric makes in his book is the difference in knowledge that the more classic manufacturing people have versus the users. The users, by definition of users, are expert on their requirements. I mean, that shouldn't be a big theme. But if somebody came here and said, no, no, the experts on the users requirement are market analysis people-- so you hire your favorite market research company, and they will tell you, no, no. Remember, market analysis, by definition, means you need information to analyze.

When you are here in the marketplace, you have information. So that's when those companies can do their job. When you are here, there is no information. The information is in the head of these leading-edge users. Which is why if you try to apply the processes that work for more mature products, where you get a lot of information and you do segmentation, you just totally miss these end users.

So the users are the ones who have the information. Now, the manufacturers have a pretty good idea about what they can build. They don't know if anybody would buy it. They have a big space they could build. I mean, let's take an example, which, again, is at the leading edge of petaflop supercomputing. I mean, if somebody here says, I want a petaflop supercomputer for a certain amount of money, and the manufacturers have no idea how to do that, that's that. You want it, but we don't know how to build it. So you have to wait.

What happens is that eventually, you have more and more information about what kinds of approaches you could take to a petaflop supercomputer. And there will be multiple approaches. And these people know the kinds of problems they want solved. And so now you start working with them to see, is what we can do something that can apply to your problems? And if you start getting a match, then all of a sudden, you start making the connection between the manufacturer and the user. Yes.

AUDIENCE: In some of the examples that you've mentioned in the last few minutes, it seems like the manufacturing and the user, especially the lead users, are actually one and the same. So if you think about Mark Zuckerberg and Facebook or Craig and Craigslist and even in the supercomputing example--

IRVING WLADAWSKY-BERGER: They were the face of the-- no, no, let's say Facebook, they were, for instance, the beginning. But then weren't there other kids at Harvard using Facebook?

AUDIENCE: There definitely were, but the manufacturers were users themselves. So Mark Zuckerberg was a student who wanted to be able to interact. And Craig wanted to sell. With the supercomputing example, you didn't have that. But you ended up opening up supercomputing source code and the infrastructure so that the users became manufacturers to some degree.

IRVING WLADAWSKY-BERGER: Exactly.

AUDIENCE: So in every example, it seems like instead of having the distinction--

IRVING WLADAWSKY-BERGER: No, exactly. In fact, that is the most critical distinction that whereas traditional you have a gap between the manufacturers and the users, in the world we're talking about, and we will discuss how big is that world, those two are much closer together than ever. And that's your point, correct?

AUDIENCE: Can it happen in a case where the manufacturer isn't the user or the user isn't able to manufacture?

IRVING WLADAWSKY-BERGER: Well, it depends what you mean by manufacture. If we include software, and if you develop a layer of open source and you encourage your users to innovate in those layers, then your users are, quote, unquote, "manufacturing"-- I mean, we can now argue, are they designing or manufacturing? But in the case of software, the difference is not very large. But no, I think the users, by this definition, have to be actively involved in helping you design and build. And each case would be a little different, how active they would be. Yes, please.

AUDIENCE: Come back to this point. I think in the case of software and the simple software, it's easy for the user to be the manufacturer. But in other cases, it may not. I might want to claim that it's very fuel-efficient, but I can't go and make it.

IRVING WLADAWSKY-BERGER: No.

AUDIENCE: The least I can do is ask somebody to.

IRVING WLADAWSKY-BERGER: But even in Facebook, after a while, the users of Facebook could require the people who had Facebook, right? So after a while, there was more of a distinction between the manufacturer and the user. Or in the case of Google, there's a lot you can do, but there is the two are distinct. It's just that you're collaborating. Yes, please.

AUDIENCE: Yeah, I mean, von Hippel gives a lot of examples that are software-based. But he also gives us the hard ones, like the surfer who modifies the board so that he straps his feet in. So actually the manufacturer had nothing to do with that. It wasn't until later that they learned. And then the whole plant thing where he took the wires. And the user was a manufacturer, but he wasn't a manufacturer of that piece. But he had a better way to use it. And I think that that happens quite a bit. At least where I come from, we get something and say, you know what? It wasn't what it is.

IRVING WLADAWSKY-BERGER: Remind me, which industry?

AUDIENCE: Well, I'm in the Coast Guard, but we get something and we say, you know what? This is OK for this, but it's really good for this. And we're using it for something totally different. And--

IRVING WLADAWSKY-BERGER: And you give the feedback back to the manufacturer?

AUDIENCE: Sometimes. [LAUGHS] But--

IRVING WLADAWSKY-BERGER: Are they searching for the feedback from you?

AUDIENCE: Not really. But sometimes if we need them to modify it on a global scale, we'll say, like at Cisco, we modified the Cisco routers to do something through RS-232 ports. And we asked them if they could do it. They said no. And we figured out how to do it. And then we went and told them. And now they sell it to me to be able to do that.

IRVING WLADAWSKY-BERGER: Did you give them the code?

AUDIENCE: Yeah. Yeah, absolutely. Because we were like, hey, if you do this, we'll buy a whole bunch.

IRVING WLADAWSKY-BERGER: No, of course. And then you don't have to support it yourself. No, no, precisely.

AUDIENCE: They modified it, and--

IRVING WLADAWSKY-BERGER: But that's a wonderful example of what we're talking about. Sung-yu, you had a point? Sung-yu is Eric's student. So he knows a lot about this area.

AUDIENCE: So I'm a little biased as to my experience. But I think the definition between the user innovation and manufacturing innovation, I have [INAUDIBLE] idea. I think it's based on how you benefit. If you benefit from innovating just by solving the problems for yourself, then there's user innovation. If you benefit by selling the innovation that you did, then it's a manufacturing innovation.

I think the point is a lot of users innovate because that benefits themselves, not by selling it. And these happen a lot in the marketplace. And because users have specific problems situation and they need to solve it and no manufacturers is supporting that problem-solving, so they have to do that for themselves. And this happens a lot. And you can see that in open source and other things as well. So I think that's the key definition. And I think that affects a lot of conceptualization.

IRVING WLADAWSKY-BERGER: Yeah. I think there's a lot more to talk about, but I want to make-- there's a point that I think is very, very important why-- I mean, we can ask ourselves, so why is this happening now? I mean, we can say, what's going on here? And I think it's a combination of factors. First of all, the, quote, unquote, "industrial age" in the Industrial economy where it's primarily about physical things, it's just harder for the users to be involved with physical things.

You can hot-rod cars. Hot rod cars are example. You take the car. You do different things. So it's not impossible. It's just much harder with physical things in an industrial economy. In a knowledge-based economy, where so much of the way you embody the function can be captured in software, which you would all agree is happening, and not just for classic software, but there's probably no discipline where that's not happening, the combination of it being software and then having the internet and collaborative technologies and the ability to distribute the software, makes it so much easier to do this distributed collaboration that would have been impossible before.

Do you all agree that that's a huge change that is industrial versus knowledge economy? And then even within, let me say, IT industry-- well, it's not just the IT industry, but in classic engineering, you tend to apply technology. You do automation with machines, pretty much, or even on back-office things. And here are the users, and you are changing things back here.

If you're a bank, you're not going to have the users of the bank help you do the transaction processing system for the bank. That would be really bizarre. So the transaction processing system is a very good example of a back-office system that supports the operations of the bank. And obviously, that's where a lot of the initial innovations happen in IT. Or medical record management-- you're not going to have your users work on the security of medical record systems. That's not what you would do.

But as we talked before, we are increasingly innovating of the stack. We are getting to the layers of problems that are services that are not industrial but are services in nature. Services, by definition, mean a person is doing something for another person. I mean, that's what a service is. And those may be employees helping each other, and maybe a doctor treating a patient. It may be somebody selling something to somebody else or giving them information.

So the services portion of the economy involves people dealing with each other. The part one hasn't focused much in engineering because we haven't had any goddamn tools to do anything with. Say, well, we'll automate this service. Well, I don't want you to automate this service. If I'm the patient, I don't want to interact with an AI doctor. I want a doctor. So he said, well, then screw you. Well, then I'll go away.

But because of the new technologies we have now, enter information analysis supercomputing, we can start bringing a whole plethora of new tools to enable the people performing the service and the people receiving this service to better work with each other. You're not automating the doctor. You're providing tools for the doctor to better serve the patient. Like, maybe tools that let them know the genomics makeup of the patient. So you can do prescriptions much better.

Or maybe also information about what treatments have worked for a community of patients. And maybe in that case, the innovation may happen between the doctor and a community of patients that had the disease and were treated who are providing information, how well they felt, how well they didn't feel. So you get this user innovation. So everything that Eric is talking about, the reason it is so profound is because I don't know which is the chicken and which is the egg.

That is, I don't know if because of tools like this and techniques like this, that's what's pushing us into the knowledge economy and allows us to bring technology and engineering to services. Or because we're in a knowledge economy and in services, we can do stuff like that. But there is something that feels very different from classic users of technology and engineering. Do you all agree with that observation. Please.

AUDIENCE: Well, one thing is user innovation may not always be what the manufacturer wants. Because, for example, if you take the example of Apple iPhone, unlocking the iPhone might be a user design innovation, but the manufacturer wouldn't want to do that. So I think user innovation is a good thing, but it might actually violate the--

IRVING WLADAWSKY-BERGER: No, I understand. I agree totally. And we don't know yet whether unlocking the phone is good for Apple or not, because you need a little time to know whether Apple refusing to-- ah, you came just in time. We'll ask Professor von Hippel this very good question you brought up. Let me first introduce you to Eric von Hippel, who wrote the book *Democratizing Innovation.*

And Eric, before you got here, I had made a comment that a critical part why user innovation is so important now is, in my opinion, two critical things are changing. One is we're moving-- please-- from an industrial economy to a more knowledge-based economy. And in a knowledge-based economy, a lot of the innovation is being captured in software, which makes it easier to do this innovation as opposed to physical things is harder, not impossible, but harder.

And the second one is that in the past, we didn't have the technologies, the tools to innovate at the services layers of businesses and economies. And by definition, services involve people doing things for each other. We were innovating in the more back-office part. You can see why John is so good. Of course, he was out there. And we had VP of marketing, and he was out there. Also, it depends on the-- the thing we are most careful with is financial analysts. Because if you say the wrong thing to financial analysts, there are even legal implications to. So you have to be careful.

IT analysts and press, the reason you have to be careful is because if you are not savvy about how your words could get interpreted, you may find reading something and [? Sadwick ?] said, I never said that. Well, but [? Sadwick, ?] maybe you said something that could have been interpreted as you said that. Also, once you do this, you become good at reading what the reporter is trying to take you to. And if you think they're trying to take you along a certain path, you start taking them along.

You need experience. And I believe that experience is very, very important. I think in the first lecture we had in class, I think one of you asked me, well, isn't that the job of marketing people-- to communicate that? Remember, we're talking about disruptive innovations, complex technology, complex systems. The people that are often the best with press and analysts are the technical people who have a really good in-depth understanding and who can best represent that.

That is, yes, you have these messages, but the really good technical people know where these messages came from, what's behind them. And they will be very, very good at reacting. And if you don't have that, you come across a little more shallow. I don't mean you're shallow. You just don't have the depth. And analysts and press, they can smell that. They can smell that, oh, so this is where I have to go.

Whereas they know that no matter where they want to go, oh, this person is really well prepared. And this person is also market savvy and articulate, particularly because he or she has thought about it. It will just be easier. So it's a very big part of the job. And the alternative is that you do wonderful work, and then you're reading the paper that your competitor is getting all the glory. And you'll say, I've been out marketing. And that is humiliating.

I mean, if somebody's going to outmarket you, let them have a better product. But there's nothing worse than you have a superb technology, a superb product, you didn't pay attention to these things. And you competitor walked away with all the glory. And you have a better technology and product, and you can write a book whining about it. But I cannot tell you, the marketplace could care less about you.