INTERVIEWS WITH THE FOUNDERS OF

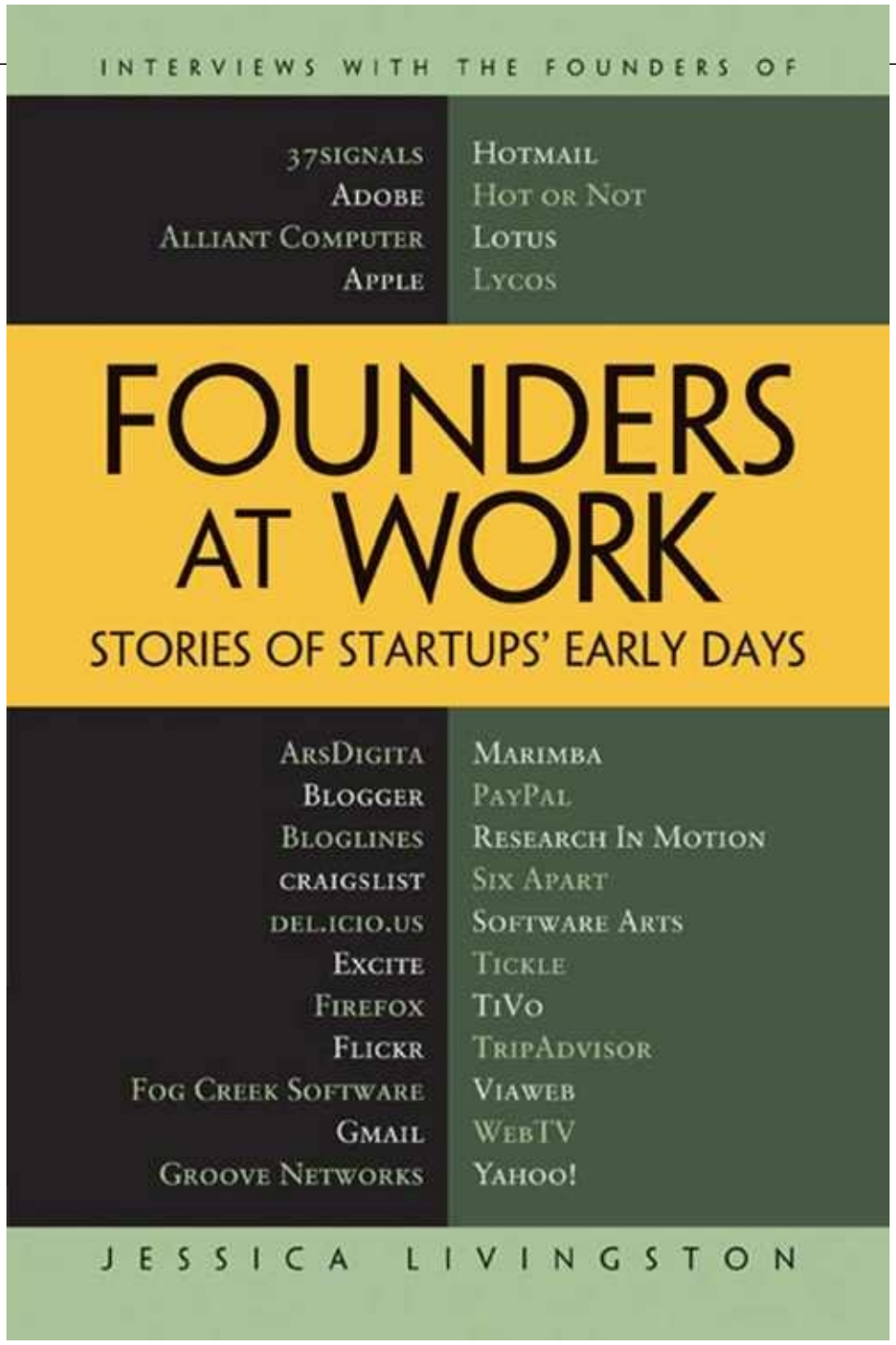
37SIGNALS
ADOBE
ALLIANT COMPUTER
APPLE
HOTMAIL
HOT OR NOT
LOTUS
LYCOS

FOUNDERS AT WORK

STORIES OF STARTUPS' EARLY DAYS

ARSDIGITA
BLOGGER
BLOGLINES
CRAIGSLIST
DEL.ICIO.US
EXCITE
FIREFOX
FLICKR
FOG CREEK SOFTWARE
GMAIL
GROOVE NETWORKS
MARIMBA
PAYPAL
RESEARCH IN MOTION
SIX APART
SOFTWARE ARTS
TICKLE
TIVO
TRIPADVISOR
VIAWEB
WEBTV
YAHOO!

JESSICA LIVINGSTON


INTERVIEWS WITH THE FOUNDERS OF
37SIGNALS
ADOBE
ALLIANT COMPUTER
APPLE
HOTMAIL
HOT OR NOT
LOTUS
LYCOS
FOUNDERS AT WORK
STORIES OF STARTUPS' EARLY DAYS
ARSDIGITA
BLOGGER
BLOGLINES
CRAIGSLIST
DEL.ICIO.US
EXCITE
FIREFOX
FLICKR
FOG CREEK SOFTWARE
GMAIL
GROOVE NETWORKS
MARIMBA
PAYPAL
RESEARCH IN MOTION
SIX APART
SOFTWARE ARTS
TICKLE
TIVO
TRIPADVISOR
VIAWEB
WEBTV
YAHOO!
JESSICA LIVINGSTON

FOUNDERS AT WORK

STORIES OF STARTUPS' EARLY DAYS

Jessica Livingston

Apress®

Lead Editor: Jim Sumser

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonatha
Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakesha
Jim Sumser, Matt Wade

Project Manager: Elizabeth Seymour

Copy Edit Manager: Nicole Flores

Copy Editor: Damon Larson

Assistant Production Director: Kari Brooks-Copony

Compositor: Dina Quan

Proofreader: Linda Seifert

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Library of Congress Cataloging-in-Publication Data

Livingston, Jessica.

Founders at work : stories of startups' early days / Jessica

Livingston.

p. cm.

ISBN 1-59059-714-1

1. New business enterprises--United States--Case studies. 2.

Electronic industries--United States--Case studies. I. Title.

HD62.5.L59 2007

658.1'1--dc22

2006101542

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Stree 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 21 Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or vis http://www.apress.com.

Contents

Foreword

Apparently sprinters reach their highest speed right out of the blocks, and spend the rest of th race slowing down. The winners slow down the least. It's that way with most startups too. Th earliest phase is usually the most productive. That's when they have the really big ideas. Imagin what Apple was like when 100% of its employees were either Steve Jobs or Steve Wozniak.

The striking thing about this phase is that it's completely different from most people's idea what business is like. If you looked in people's heads (or stock photo collections) for imag representing "business," you'd get images of people dressed up in suits, groups sitting arou conference tables looking serious, Powerpoint presentations, people producing thick reports f one another to read. Early stage startups are the exact opposite of this. And yet they're probably th most productive part of the whole economy.

Why the disconnect? I think there's a general principle at work here: the less energy peop expend on performance, the more they expend on appearances to compensate. More often than n the energy they expend on seeming impressive makes their actual performance worse. A few yea ago I read an article in which a car magazine modified the "sports" model of some production c to get the fastest possible standing quarter mile. You know how they did it? They cut off all th crap the manufacturer had bolted onto the car to make it *look* fast.

Business is broken the same way that car was. The effort that goes into looking productive is n merely wasted, but actually makes organizations less productive. Suits, for example. Suits do n help people to think better. I bet most executives at big companies do their best thinking whe they wake up on Sunday morning and go downstairs in their bathrobe to make a cup of coffe That's when you have ideas. Just imagine what a company would be like if people could think th well at work. People do in startups, at least some of the time. (Half the time you're in a pan because your servers are on fire, but the other half you're thinking as deeply as most people on get to sitting alone on a Sunday morning.)

Ditto for most of the other differences between startups and what passes for productivity in b companies. And yet conventional ideas of "professionalism" have such an iron grip on our mind that even startup founders are affected by them. In our startup, when outsiders came to visit w tried hard to seem "professional." We'd clean up our offices, wear better clothes, try to arrange th a lot of people were there during conventional office hours. In fact, programming didn't get don by well-dressed people at clean desks during office hours. It got done by badly dressed people was notorious for programming wearing just a towel) in offices strewn with junk at 2 in th morning. But no visitor would understand that. Not even investors, who are supposed to be able recognize real productivity when they see it. Even we were affected by the conventional wisdom

We thought of ourselves as impostors, succeeding despite being totally unprofessional. It was as we'd created a Formula 1 car but felt sheepish because it didn't look like a car was supposed look.

In the car world, there are at least some people who know that a high performance car looks li a Formula 1 racecar, not a sedan with giant rims and a fake spoiler bolted to the trunk. Why not business? Probably because startups are so small. The really dramatic growth happens when startup only has three or four people, so only three or four people see that, whereas tens thousands see business as it's practiced by Boeing or Philip Morris.

This book can help fix that problem, by showing everyone what, till now, only a handful peop got to see: what happens in the first year of a startup. This is what real productivity looks lik This is the Formula 1 racecar. It looks weird, but it goes fast.

Of course, big companies won't be able to do everything these startups do. In big compani there's always going to be more politics, and less scope for individual decisions. But seeing wh startups are really like will at least show other organizations what to aim for. The time may soo be coming when instead of startups trying to seem more corporate, corporations will try to see more like startups. That would be a good thing.

Paul Graha

Acknowledgments

I'd first like to thank my aunt, Ann Gregg, for her unfailing support and encouragement. She an extraordinarily perceptive reader and she provided a lot of advice that helped make this a bett book.

Thanks to the people I interviewed for sharing their stories and their time. One thing I noticed the interviews that I didn't mention in the introduction is how much I *liked* the founders. The were genuine and smart, and it was an honor to talk with them. I know the candid nature of the stories and advice will inspire would-be founders for years to come.

Thanks to Gary Cornell for being willing to do a different kind of book, and to the Apress tea for working on a different kind of book.

I'd like to thank many people for their willingness to make introductions: Jim Baum, Patri Chung, Mark Coker, Jay Corscadden, Rael Dornfest, Jed Dorsheimer, Randy Farmer, Stev Frankel, Anand Gohel, Laurie Glass, James Hong, Mitch Kapor, Morgan Ley, Mike Palmer, Tor Palmer, Bryan Pearce, Andrew Pojani, Will Price, Ryan Singel, Langley Steinert, Chris Sacca, an Zak Stone.

Thanks to Kate Courteau for creating cozy offices for me to work in; Lesley Hathaway for all h advice and support; Alaina and David Sloo for their many introductions; and Sam Altman, Pa Buchheit, Lynn Harris, Marc Hedlund, and Aaron Swartz, who read early chapters of the book. owe thanks to Lisa Abdalla, Michele Baer, Jen Barron, Ingrid Bassett, Jamie Cahill, Jessi Catino, Alicia Collins, Caitlin Crowe, Julie Ellenbogen, John Gregg, Chrissy Hathaway, Kat Helmer, Susan Livingston, Nadine Miller, Sara Morrison, Bridget O'Brien, Becky Osborn Allison Pellegrino, Jennifer Stevens, and Suzanne Woodard for their encouragement.

Thanks to others who shared their insights on startups at Y Combinator dinners or with m personally: Rich Bacon, Greg Benning, Tom Churchill, Michael Ellenbogen, Jonathan Gertle Hutch Fishman, Sara Harrington, Bill Herp, Bradley Horowitz, Joel Lehrer, Carolynn Levy, Simo London, Page Mailliard, Udi Manber, Fredrick Marckini, Greg McAdoo, Mark Macenka, Mil Mandel, Jerry Michael, Rich Miner, Mark Nitzberg, Peter Norvig, Steve Papa, Tom Pinckney, Sta Reiss, Olin Shivers, Hugues Steinier, Jeff Taylor, Rob Tosti, and Stephen Wolfram.

Thanks to the founders of all the startups we've funded at Y Combinator. They are inspiration and I know they will have valuable stories of their own to share.

Special thanks to Trevor Blackwell and Robert Morris for all of their support. I'm lucky to wo
with them.

To my grandparents, Baba and Bob, who I admire and whose advice from their own experienc
as authors helped me a lot. Extra special thanks to Dad and Michele, who supported me even wh
I had crazy ideas like quitting my job to start a company and work on a book. Over the years, m
father never seemed to doubt that I could do something I'd be really proud of, and I'm ve
appreciative.

Most of all, thanks to Paul Graham. He inspired this book and was a source of encouragem
and advice throughout the entire process. I'm grateful to have benefited from his extraordina
understanding of technology, startups, and writing. But mostly, I'm glad to know him.

Introduction

Some kind of magic happens in startups, especially at the very beginning, but the only peop there to see it are the founders. The best way to understand what happens is to ask them, so tha what I did.

In this book, you'll hear the founders' stories in their own words. Here, I want to share some the patterns I noticed. When you're interviewing a series of famous startup founders, you can't he trying to see if there is some special quality they all have in common that made them succeed.

What surprised me most was how unsure the founders seemed to be that they were actually on something big. Some of these companies got started almost by accident. The world thinks startup founders as having some kind of superhuman confidence, but a lot of them were uncerta at first about starting a company. What they weren't uncertain about was making something goo —or trying to fix something broken.

They all were determined to build things that worked. In fact, I'd say determination is the sing most important quality in a startup founder. If the founders I spoke with were superhuman in an way, it was in their perseverance. That came up over and over in the interviews.

Perseverance is important because, in a startup, nothing goes according to plan. Founders liv day to day with a sense of uncertainty, isolation, and sometimes lack of progress. Plus, startups, b their nature, are doing new things—and when you do new things, people often reject you.

That was the second most surprising thing I learned from these interviews: how often th founders were rejected early on. By investors, journalists, established companies—they got th Heisman from everyone. People like the idea of innovation in the abstract, but when you prese them with any specific innovation, they tend to reject it because it doesn't fit with what the already know.

Innovations seem inevitable in retrospect, but at the time it's an uphill battle. It's curious to thin that the technology we take for granted now, like web-based email, was once dismissed a unpromising. As Howard Aiken said, "Don't worry about people stealing your ideas. If your idea are any good, you'll have to ram them down people's throats."

In addition to perseverance, founders need to be adaptable. Not only because it takes a certa level of mental flexibility to understand what users want, but because the plan will probab change. People think startups grow out of some brilliant initial idea like a plant from a seed. B

almost all the founders I interviewed changed their ideas as they developed them. PayPal start
out writing encryption software, Excite started as a database search company, and Flickr grew o
of an online game.

Starting a startup is a process of trial and error. What guided the founders through this proce
was their empathy for the users. They never lost sight of making things that people would want.

Successful startup founders typically get rich from the process, but the ones I interviewe
weren't in it just for the money. They had a lot of pride in craftsmanship. And they wanted
change the world. That's why most have gone on to new projects that are just as ambitious. Sur
they're pleased to have more financial freedom, but the way they choose to use it is to kee
building more things.

Startups are different from established companies—almost astonishingly so when they are fir
getting started. It would be good if people paid more attention to this important but ofte
misunderstood niche of the business world, because it's here that you see the essence
productivity. In its plain form, productivity looks so weird that it seems to a lot of people to b
"unbusinesslike." But if early-stage startups are unbusinesslike, then the corporate world might b
more productive if it were less businesslike.

My goal with these interviews was to establish a fund of experience that everyone can lea
from. You'll notice certain classes of problems that constantly bit people. All the founders ha
things they wished they'd known when they were getting started. Now these are captured for futu
founders.

I'm especially hoping this book inspires people who want to start startups. The fame that com
with success makes startup founders seem like they're a breed apart. Perhaps if people can see ho
these companies actually started, it will be less daunting for them to envision starting somethi
of their own. I hope a lot of the people who read these stories will think, "Hey, these guys we
once just like me. Maybe I could do it too."

CHAPTER 1

Max Levchin
Cofounder, PayPal

PayPal was founded in December 1998 by recent college grad Ma Levchin and hedge fund manager Peter Thiel. The company went through several ideas, includin cryptography software and a service for transmitting money via PDAs, before finding its niche as web-based payment system. That service became wildly popular for online vendors, especial eBay sellers, who preferred it to traditional payment methods. PayPal went public in early 200 and was acquired later that year by eBay for $1.5 billion.

PayPal was started during the Internet Bubble, but it was in no sense a Bubble startup. I success was a direct reflection of the intelligence of the people who built it. PayPal won becaus they built a better mousetrap.

With any new method of moving money comes new forms of fraud. In large part, PayP succeeded because it could deal with fraud—and its competitors couldn't. The software th Levchin and his team developed to combat fraud runs quietly and invisibly. To this day, PayP doesn't talk much about it. But Levchin's software was just as much the reason for PayPal's succe as a more visible product like the Apple II was for Apple.

Livingston: Tell me a little about how PayPal got started.

Levchin: The company was really not founded to do payments at all. My focus in college w security. I wanted to do crypto and stuff like that. I had already founded three different compani during college and the year after, which I spent in Champaign-Urbana, where I went to schoo Then, in favor of not doing graduate school, I decided to move out to Silicon Valley and try to sta

another company.

So I was hanging around Silicon Valley in the summer of '98 and was not really sure what I wa going to do with my life. I was living in Palo Alto, squatting on the floor of a friend. I went to s this random lecture at Stanford—given by a guy named Peter, who I had heard about, but nev met before.

The lecture turned out to have only six people in it. It was in the heat of the summer, so nobo showed up. This guy was like, "There are only six of you, OK." Afterwards I walked up to talk him. He was this really intense guy, and he said, "We should get breakfast sometime." So we m up the next week.

I had two different ideas that I was considering starting companies around, and I pitched him both evenly. Peter was running a hedge fund at the time. For a few weeks we kept talking, a eventually he said, "Take this idea, because this one is better, and you go start a company arou it, and then I can have my hedge fund invest a little bit of money in it"—like a couple hundr thousand dollars. That was a good thing, since I was starting to run out of money.

I had just moved from Champaign; most of my contacts and friends were in Chicago. One them I was trying to convince to be the CEO. He wasn't really available, so I wound up bei without a CEO. I called Peter and said, "This investment is a great thing, but I have no one to r the company. I'm just going to write the code and recruit the coders." And he said, "Maybe I cou be your CEO." So I said, "That's a really good idea." The next 2 weeks we were sort of playi with the idea, and by 1/1/99 we agreed that he would be the CEO and I would be the CTO.

Livingston: How did you have the idea?

Levchin: The initial idea was actually very different. At the time, I was really into develop software for handheld devices, which is sort of an art and a science unto its own. And I was rea into security. This idea that I had in college, which I was vaguely successful with—if you've ev seen these authentication devices, like a little card that spits out numbers at you that you can log with. It's like a one-time password generator, like S/Key, Digital Pathways, and CRYPTOCar Most of the algorithms are variations on the standard called X9.9, which is a public standard. T algorithms don't really use it correctly. In college one day I had bought all the different kinds cards. Each costs like $50 or $100, so it's not that expensive. They weren't that difficult to revers engineer because you already know the standard, so you know it can't be too far outside t standard. I reverse-engineered most of them except for one which was very proprietary. I decid not to touch that one since I was too poor to handle a lawsuit.

Once I got them all reverse-engineered, I wrote an emulator for every single type of them fo Palm Pilot. I had a lot of friends on campus who were really into security as well—most of the were sys admins—and they carried a whole bunch of these things in their pockets, because most the time you can only use one per computer, per system. If you adminned a lab with ten serve

you'd have a stack of these things in your pocket, and that adds up. They are heavy, and they nee batteries. I basically emulated the whole thing on a Palm Pilot so my friends were able to thro out their stupid devices and use my thing.

I posted it on the Web, which was young and silly then, and I got hundreds and then thousands downloads, and people were offering me money to get more features in. So I thought, "This seem to be a business." At the time, I was just keen on getting any sort of business off the ground. S when I moved to the Valley, I basically pitched Peter on the following concept. There's clearl demand for moving these cryptographic operations that are poorly understood. Even though it's n rocket science to reverse-engineer this stuff, no one else had done it before me, so there's som complexity involved.

The real difficult thing actually was getting an implementation of a cryptographic algorithm on Palm Pilot, because Palm Pilots are very low power, and, back then, they were *really* low power- like a 16 MHz processor. So, to do an encryption of a public key operation on a Palm Pilot wa really expensive. There is some art involved in how you speed it up—both from the user interfa perspective and the math perspective. In math, you have to see how much you can squeeze out it, and in the user interface, you have to make it feel like it's not taking that long, even though really is taking like 2 seconds, which is a really long time.

On these handheld devices, the cards that you get, you type in the password and it's done. I wa able to get it to the point where it was instantaneous on a Palm Pilot. These things are all sort child's play at this point, but at the time they were very important. Anyway, I wanted to start company that would take this scarce skill of implementing crypto on handheld devices and the packaging it into libraries and products. The assumption was that the enterprises are going to a go to handheld devices really soon as the primary means of communication. Every corporate do in America will hang around with a Palm Pilot or some kind of a device. What I wanted to do wa capitalize on that emergence of technology. And then, of course, enterprise requires security security requires these scarce skills; I have the skills; start a company.

So that's what Peter funded. By the time he joined, we had realized that, even though the theo was pretty much logical, the move of the enterprise to handheld devices was actually n forthcoming. Kind of like the early Christians in the first century were all really hard at wo waiting for the second coming. Still waiting. So it felt like the early Christians. "Any minute nov there'll be millions of people begging for security on their handheld devices." It just wasn happening. We were correct to change our strategy, since it still hasn't happened.

Livingston: Tell me about how you adapted the strategy.

Levchin: Initially, I wanted to do crypto libraries, since I was a freshly minted academic. won't even need to figure out how to do this commercialization part. I'm just going to bui libraries, sell it to somebody who is going to build software, and I can just sit there and make penny per copy and get marvelously rich very quickly." But no one was making the softwa

because there was no demand. So we said, "We'll make the software." We went to enterprises an told them we were going to do this and got some positive reception, but then the thing happen again where no one really wants the stuff. It's really cool, it's mathematically complex, it's ve secure, but no one really needed it.

By then we had built all this tech that was complicated and difficult to understand and replicat so we thought, "We have all these libraries that allow you to secure anything on handheld device What can we secure? Maybe we can secure some consumer stuff. So enterprises will go away, a we'll go to consumers. We'll build the wallet application—something that can store all of yo private data on your handheld device. So your credit card information, this and that." And we di and it was very simple because we already had all the crypto stuff figured out. But, of course, the was no incentive to have a wallet with all these digital items that you couldn't apply anywher "What's my credit card number?" Pull out your wallet and look, or pull out your handheld wall and look? So that was really not going to happen either.

Then we started experimenting with the question: "What can we store inside the Palm Pilot th *is* actually meaningful?" So the next iteration was that we'd store things that were of value and yo wouldn't store in other ways. For example, storing passwords in your wallet is a really bad idea. you store them in your Palm Pilot, you can secure it further with a secondary passphrase th protects it. So we did that, and it was getting a little bit of attention, but it was still very amateur.

Then finally we hit on this idea of, "Why don't we just store *money* in the handheld devices The next iteration was this thing that would do cryptographically secure IOU notes. I would say, owe you $10," and put in my passphrase. It wasn't really packaged at the user interface level as a IOU, but that's what it effectively was. Then I could beam it to you, using the infrared on a Pal Pilot, which at this point is very quaint and silly since, clearly, what would you rather do, take o $5 and give someone their lunch share, or pull out two Palm Pilots and geek out at the table? B that actually is what moved the needle, because it was so weird and so innovative. The geek crow was like, "Wow. This is the future. We want to go to the future. Take us there." So we got all thi attention and were able to raise funding on that story.

Then we had the famous Buck's beaming—at Buck's restaurant in Woodside, which is sort of th home away from home for many VCs. Our first round of financing was actually transferred to u via Palm Pilot. Our VCs showed up with a $4.5 million preloaded Palm Pilot, and they beamed to us.

The product wasn't really finished, and about a week before the beaming at Buck's I realized th we weren't going to be able to do it, because the code wasn't done. Obviously it was really simp to mock it up—to sort of go, "Beep! Money is received." But I was so disgusted with the idea. W have this security company; how could I possibly use a mock-up for something worth $4 million? What if it crashes? What if it shows something? I'll have to go and commit ritual suici to avoid any sort of embarrassment. So instead of just getting the mock-up done and gettin reasonable rest, my two coders and I coded nonstop for 5 days. I think some people slept; I know didn't sleep at all. It was just this insane marathon where we were like, "We have to get this thin

working." It actually wound up working perfectly. The beaming was at 10:00 a.m.; we were don
at 9:00 a.m.

It was one of these things where you can't just be done. With crypto, if you are one bit of
nothing's going to work. We started testing at midnight the night before and fixed all the bugs an
tested more. There were definitely some memory leaks, but it was secure. It was one of the
things where the software wasn't perfect, but the security path where the money changed hand
was definitely provably secure. The danger was that the Palm Pilots might crash, but th
transaction was perfectly safe. I could have bet my own life on the transaction. The thing that w
not safe was just the software was not really perfect. It was clunky; I was worried that it mig
crash.

So we had stacks and stacks of Palm Pilots preloaded with the same software. Obviously, mone
could only reside in one of them, but the plan was that, if I see that any one of them is crashin
I'm going to make a fresh pair, because we needed two Palm Pilots, one for the receiving and on
for the sending. I was fully prepared. They were marked, "Sender A, Sender B, Sender C, Receiv
A, Receiver B, Receiver C." So I had this stack of Palm Pilots, I hopped in a car, drove to Buck'
and it was like 9:50 a.m. Peter was getting very anxious about the whole thing. That's whe
everything becomes very blurry, because I was so tired by then.

There were about a dozen TV cameras and journalists—there was really big coverage. We di
the beaming, and some group showed up late and said, "Well, can you do it again?" I said, "No,
just slaved away for 5 days straight—for 5 months straight. The whole point of the security is th
you can't replicate the transaction. Once it's done, the money has changed hands." So these guy
actually made Peter pretend like it was going to happen and turned away the screen—because th
screen was actually saying, "Security breach! Don't try to resend the same money again." Whic
was a triumph for me, but a pain in the ass for the camera.

As I was getting interviewed by the *Wall Street Journal*, or some big pub guy, all I rememb
was that he went off to the bathroom for a second, and they brought out my omelet. The next thin
I remember, I woke up, and I was on the side of my own omelet, and there was no one at Buck'
Everyone was gone. They just let me sleep.

Livingston: What did you do first after you got this new funding?

Levchin: As soon as we got funding, we started hiring aggressively, and we built this app fo
the Palm Pilot, which was getting pretty good growth. We were getting 300 users a day. Then w
built a demo for the website, which was functional, so you could do everything on the website th
you could do on a Palm Pilot, except the website was unsexy and we didn't really care. It was lik
"Go to the website and download the Palm Pilot version. It's really cool."

Livingston: Three hundred people were downloading it per day? For fun?

Levchin: Well, there are lots of geeks. It slowed down pretty quickly too, but initially we got lot of publicity about it.

Sometime by early 2000, we realized that all these people were trying to use the website f transactions, and the growth of that was actually more impressive than the growth of the handhe device one, which was inexplicable, because the handheld device one was cool and the website w just a demo. Then all these people from a site called eBay were contacting us and saying, "Can put your logo in my auction?" And we were like, "Why?" So we told them, "No. Don't do it." S for a while we were fighting, tooth and nail, crazy eBay people: "Go away, we don't want you."

Eventually we realized that these guys were begging to be our users. We had the moment o epiphany, and for the next 12 months just iterated like crazy on the website version of the produc which is today's PayPal. Sometime by late 2000, we killed the handheld one because we peake out at 12,000 users. They were still using it a little bit, and they were really upset when we kille it. They said, "You were about the handheld transactions, not about this web stuff." We're like "No, we're pretty much about the web stuff."

Livingston: How many users did you have for the website when you killed the handhe product?

Levchin: I think we must have been 1.2 . . . 1.5 million users. It was an emotional b completely obvious business decision.

Livingston: When did you first notice fraudulent behavior?

Levchin: From day one. It was pretty funny because we met with all these people in the bankin and credit card processing industry, and they said, "Fraud is going to eat you for lunch." We said "What fraud?" They said, "You'll see, you'll see."

I actually had an advisor or two from the financial industry, and they said, "Get ready fo chargebacks. You need to have some processing in place." We said, "Uh huh." They said, "You don't know what a chargeback is, do you?"

Livingston: So you didn't foresee this fraud?

Levchin: I had no idea what was going to happen.

Livingston: But you weren't too surprised?

Levchin: We tried to attack the system for ourselves, like a good security person would. How can you cheat and steal money and do whatever? We made some provisions from day one t prevent fraud. We prevented all the obvious fraud, and then, I think 6 months into it, we saw th

first chargeback and were like, "Ah, one per week. OK." Then it was like an avalanche of losse
2000 was basically the year of fraud, where we were just losing more and more and more mone
every month. At one point we were losing over $10 million per month in fraud. It was crazy.

That was when I decided that that was going to be my next challenge. I started researching
figuring out what could be done and attacking the problem.

Livingston: So you made a conscious decision to attack this problem?

Levchin: It was actually sort of a side effect. We had this merger with a company called X.co
It was a bit of a tough merger because the companies were really competitive—we were two lar
competitors in the same market. For a while, Peter took some time off. The guy who ran X.co
became the CEO, and I remained the CTO. He was really into Windows, and I was really in
Unix. So there was this bad blood for a while between the engineering teams. He was convinc
that Windows was where it's at and that we have to switch to Windows, but the platform that w
used was, I thought, built really well and I wanted to keep it. I wanted to stay on Unix.

By summer 2000, it seemed like the Windows thing was going to happen because Peter wa
gone. He took a sabbatical to make sure there were no clashes between the CEOs. So, this oth
guy was pushing me toward accepting that Windows was going to be the platform. I said, "Well,
this is really going to happen, I'm not going to be able to provide much value, because I don
really know anything about Windows. I went to a school that was all Unix all the time, and I spe
all my life coding for Unix."

I had this intern that I hired before the merger, and we thought, "We built all these cool Uni
projects, but it's kind of pointless now because they are going to scrap the platform. We might a
well do something else." So he and I decided we were going to find ourselves fun projects. We di
one kind of mean project where we built a load tester package that would beat up on the Windo
prototype (the next version was going to be in Windows). We built a load tester that would te
against the Unix platform and the new Windows one and show in beautiful graphs that th
Windows version had 1 percent of the scalability of the Unix one. "Do you really want to do that?

It was me acting out, but it was kind of a low time for me because I was not happy with the wa
we were going. Part of having a CEO is that you can respectfully disagree, but you can resign
you don't like it that much.

But then eventually I became interested in the economics of PayPal and trying to see what
going on in the back end, because I was getting distracted from code and technology. I realize
that we were losing a lot more money in fraud than I thought we were. It was still early 2001.
you looked at the actual loss rates, they were fairly low. You could see that we were losing money
but, given the growth of the system and the growth of the fraud, fraud was not that big of
problem. It was less than 1 percent—it was really low. But then, if you looked at the rate of grow
of fraud, you could see that, if you don't stop it, it would become 5 percent, 10 percent of th

system, which would have been prohibitive.

So I started freaking out over it, and this intern and I wrote all sorts of packages—very statistic stuff—to analyze "How did it happen; how do we lose money?" By the end of the summer, v thought, "The world is going to end any minute now." It was obvious that we were really losi tons of money. By mid-summer, it was already on a $10 million range per month and just ver scary.

Livingston: Did the rest of the company know you were right?

Levchin: Through the summer, I think various people were slowly coming to understand th this thing was really serious. It was pretty obvious at a certain point. I didn't have to real convince anyone. In the beginning some people said, "Yes, it's a lot of money, but we're reall growing, too. As an absolute amount, $5 million is a lot of losses, but, if you are processing $30 million, whatever."

There was actually a bit of an altercation at the very top management level, which caused th CEO to leave. Peter came back as the CEO. The first decision that he and I took was that my ne job—in addition to technology—was going to be this fraud thing, because I already spent so muc time looking at it. This guy Bob, the intern, and I—I convinced him to drop out of Stanford for year and work with me more on it—for the next year, we just worked nonstop on trying t understand and fix these problems.

Livingston: So the CEO left and Peter came back?

Levchin: The three of us are pretty good friends now. At the time, already I had hated the guy guts for forcing me to do Windows, and then, in the end, I was like, "You gotta go, man." M whole argument to him was, "We can't switch to Windows now. This fraud thing is most importan to the company. You can't allow any additional changes. It's one of these things where you want t change one big thing at a time, and the fraud is a pretty big thing. So introducing a new platform doing anything major—you just don't want to do it right now." That was sort of the trigger for fairly substantial conflict that resulted in him leaving and Peter coming back and me taking ov fraud.

Livingston: When was the first time that you said, "This is working"?

Levchin: Bob and I built this package called IGOR. We had all these different things that wer all named after various Russian names—and they had to be four characters long and start with a *I*. It was sort of a random requirement that I came up with. We had IGOR, INGA, IVAN—at leas two more. So we built this tool—actually we have a patent on it now—and it was very impressiv It's based on the assumption of all sorts of convoluted guesses on our part, but the guesses turn o to be mostly right.

We actually had these human investigators, like 20 to 30 human investigators, that would try t unravel particularly large fraud cases and see if we could recover some money or send the Fe after somebody. We didn't really have much success sending people after criminals. All they'd tr to do is see where the money went and see if we could recover some of it before it left the syster That was pretty difficult to do because the tools we had available to us at the time allowed you look at only a couple of accounts at the same time. If you had a well-coordinated fraud, wi thousands of accounts or hundreds of thousands of accounts involved, you basically didn't kno how to follow it.

I remember walking into the cubicle of one of the investigators, and he had volumes ar volumes of printouts. I asked what it all was, and he said, "I'm tracing some money." I said, "Ho many cases is this?" And he said, "This is just one case." I said, "How much money are we talkir about?" He said, "It's like $80,000 worth of losses." "Well, that's a lot of money, but it's taken yo clearly at least a week to print this stuff out."

We realized that the way we were attacking these things was just fundamentally flawed. So Bo and I built this system that was part visualization package, part graph balancing tool, that wou try to represent large-scale travels of money in the system in a visual form. Taking that as a bas we built all these different tools that would allow computers to predict where particular expensive losses would be and then represent the networks of losses to the investigators in such way that they could very quickly make a decision whether or not to pursue a particular case.

Once we had that, I sort of had this tearful moment with one of the investigators where she w just crying in happiness—"You don't even understand what you did, Max"—when we showed it t them. They were really over-worked.

Once that happened, there was this huge reduction. It wasn't like 80 percent or anything. But, a this time, we had all these different ideas and we'd bring the fraud down one-tenth of a percent one-fifth of a percent, but it was really not noticeable. Then, one day, we brought the fraud dow with that tool, a lot. So we're clearly getting better at this.

Then a woman named Sarah Imbach went into a sort of self-initiated exile. She moved to Omal and first became the manager of the fraud group and then eventually became the manager of th whole center. When the fraud group operations moved to Omaha, that made it a lot cheaper for u to run. She was working on the human management part—all the investigators—and I would l supplying her with software. Between those things, we got fraud pretty well under control in abo a year.

Livingston: So the fraud solution was a combination of humans and software?

Levchin: Depending on who you ask. I think Sarah feels that it's probably more humans and th coders think it's more technology. It's one of those things where, in the end, fraud is s nondeterministic that you need a human or a quantum computer to look at it and sort of make

final decision, because, in the end, it's people's money. You don't really want some compute saying, "$2.00 for you, nothing for you." You need a human with a brain to say, "Hmm. This look like fraud, but I really don't think it is."

Then there are various processes and exception handling where you say, "Even though it's frau you don't handle it because . . ." We got really good at it later on. Initially, we sorted things b loss, but then we started sorting things by expected loss. We'd estimate the probability of losse programmatically, and then we'd get the amount of money in question calculated, figure out th expected loss, and then sort the cases for the investigators by expected loss.

The investigators would only have to deal with the top 5 percent. You'd never go through th entire queue of things for them to judge, but, because they judge things pretty quickly, they woul go through half the queue, and they would inevitably start with the ones we thought were th highest possible loss. So, the highest probable, the highest possible. That was one of the techniqu that we used to guide development.

Livingston: Were any of your competitors doing anything similar?

Levchin: We kept the stuff under wraps for a very long time. We never really showed IGOR t anyone. We never talked about it in the press. I was definitely very paranoid. Initially, when w built it, we had a conference room where there was the IGOR terminal, and people would go there, use it, and leave. There were no other copies available.

Eventually, various federal and state authorities wanted to use it too, because they started to se that we were getting pretty good at this stuff. We would invite them in, and they would have to g into the room and use it and leave. They couldn't take it with them, couldn't print.

Livingston: Did you patent this technique?

Levchin: I didn't really want to patent it because, for one, I don't like software patents, an two, if you patent it, you make it public. Even if you don't know someone's infringing, they wi still be getting the benefit. Instead, we just chose to keep it a trade secret and not show it anyone.

After a while, IGOR became well known to the company, like all the other tools that we had bui early on. We had patented some of it, and some of it we said, "OK, it's open for wide use now. There's still a whole bunch of tools that they are using today that are not public. They don't ta about it much at all, and I think that's a good thing.

Livingston: So is PayPal in a sense a security company?

Levchin: I think a good way to describe PayPal is: a security company pretending to be