Oral History of Andy Grignon

Interviewed by:
Hansen Hsu
Marc Weber

Recorded April 18, 2017
Mountain View, CA

CHM Reference number: X8181.2017

Hsu: So, today is April 18th, 2017. I am Hansen Hsu. Next to me is Marc Weber and we are interviewing Andy Grig—

Grignon: Grignon [pronounces 'Grinyon']

Hsu: OK, Grignon is the correct pronunciation?

Grignon: I like to say it's like the steak: a filet mignon.

Hsu: Yes.

Grignon: You know, Andy Grignon.

Hsu: Okay. So, you do use the French pronunciation.

Grignon: Well, no. I grew up in Wisconsin. So, <laughter> we butcher all the names.

<laughter>

Hsu: So, Andy Grignon. So, normally, we start with your childhood, but we have a smaller amount of time today. And so, I'll just start with asking about how you first got into computers. What was your first experience with computers?

Grignon: My first experience with computers—you know, my first computer experience would've been an Atari 2600. <laughter> And I actually count those as computers. But if you were to go by the conventional sense, it would've been a Commodore 64. And I got it because my school had one and I think I was in third grade or so, and initially it was BASIC, because you still had to, like, type and program even as basic as it was. But that's what kind of led me into programming. And back in those days there was a magazine called *Compute* and if you wanted to have software that—'cause I didn't have a hard drive, well, certainly not a hard drive, but we had floppy drives and things—you would type in page after page of stuff from a magazine. Like, data, you know, like three digits, comma, you know, data. And you would do that for pages and pages and then at the end of it, assuming you didn't have any errors, it would just work. And then if somebody bumped the power cord, whatever, it was all gone, 'cause I didn't have the cassette tape or, whatever, the floppy drive to record it on. But, yeah, that was my intro to it. So, I went from a

Commodore 64, Commodore 128, got an Amiga 512—wasn't cool enough for the original Amiga, and then I got into the Apple line. So, I had a Mac and all that.

Hsu: What, what was your first Mac model?

Grignon: It was a Mac Classic is the one that I owned—and that was when I went to college. I went to the University of Iowa and it seems Apple at the time had a really great discount program for university. And so, I got this Mac Classic for nothing. I mean, it was, like, a couple hundred bucks, five hundred bucks, whatever it was.

Hsu: Wow.

Grignon: It wasn't that much money. And then they introduced the AV line. So, I had a Centra [Centris] 660 AV, I think?

Hsu: Yeah.

Grignon: And that was kind of the first intro to the PowerPC, but, you know, from then on—previously, obviously, was Motorola. [Actually this is incorrect. The Centris 660AV used a Motorola 68040 CPU, while the Power Macintosh 6100 used a PowerPC 601, but came in the same case as the 660AV. –HH] But, yeah, those are the intro computers. No, I'm sorry; before that, I had a Mac IIsi.

Hsu: Oh, I had a Mac IIsi.

Grignon: IIsi, but did you have the FPGA—or, I'm sorry, the FPU card?

Hsu: Oh, I don't think I had the floating point card.

Grignon: So, the FPU you needed in order to run A/UX, and the IIsi—[A/UX] which was Apple's UNIX, right—and IIsi was the lowest grade Mac— IIsi with the FPU daughter card could run A/UX.

Hsu: Oh.

Grignon: And I had that.

Weber: Did you ever run Intermedia?

<laughter>

Grignon: No!

Weber: That was A/UX only.

<laughter>

Grignon: No, I never—you know, A/UX was a funny—but that was, I think, my first fascination with UNIX, right? I mean it was based on System V [pronounced "Five"] and in those early days that's when we were—you know, System V was kind of the gateway hacker OS, right? If you wanted—like the early days of script kitties. If you wanted to get into hacking phone systems or blue boxes is what I built and those were the early days of kind of hacking. And so, System V UNIX was my thing. I was like, "Oh, I'm gonna learn this, so I can be a hacker?" <laughs> You know, looking back it didn't make any sense knowing what you know today, but that was the idea was to have this IIsi and A/UX.

Weber: So, blue box phone stuff—?

Grignon: Blue boxing? I actually—when I went to Iowa I built a red box, which was—it was based on a Radio Shack memory dialer. So, it had a little speaker on it and you could type numbers and you would—like, "Oh, I gotta find Joe Schmoe's name," whatever, you know, and <mimics sound of ascending tones> "Boop, boop, boop"—you know, make some noises, right? Tones. Well, if you changed the crystal on this whatever, Radio Shack dialer, the star key—if you did five stars, mimicked exactly the tone that a quarter made in a pay phone. And so, that was the early days of red boxing. So, you'd have to put, like, a nickel or something in the machine at the dorm and then I could—you'd hit the button a few times and it would mimic the tones to make a quarter, and that's how I called home in those early days.

<laughter>

Grignon: Saved myself upwards of ten dollars on phone calls!

<laughter>

Grignon: But the twenty-five-dollar dialer was worth it.

<laughter>

Grignon: 'Cause that was '91. That was just kind of, you know, for a sense of time, that was in '91 is when I went to college. And so, all of this, you know, the Mac IIsi, all that, was kind of happening around that time.

Hsu: Okay.

Grignon: Yeah, yeah.

Hsu: So, are you from Iowa? Is that why you went to Iowa State?

Grignon: No, I went to—like, I was born in Wisconsin, lived in Houston and ultimately grew up in the Chicago suburbs, a suburb called Lake Forest. And I went to Iowa because, originally, I wanted to be a doctor. And I had been programming computers, you know, ever since I was a kid and for whatever reason I got it in my head that to be successful you needed to go into traditional things like law or medicine or—and Iowa actually was a pretty good medical school. But I got weeded out, 'cause I failed chemistry. It wasn't even the hard chemistry. It wasn't even organic. I failed basic chemistry. And I was, like, "Well, I'm not going to be a good doctor if I can't really pass chemistry." And so, I went back to what I loved, which was programming, and it turns out computer science—Iowa wasn't at the time an accredited university for Comp Sci. And so, I—but I started taking classes and I really enjoyed it. So, when I was in high school, we had a programming class called Fortran and—Fortran, obviously, a language from way back when. Mr. Fassell [ph?] was my teacher. Odd, the weird things you remember, right? But my one kind of brush with fame was in Mr. Fassell's Fortran class the guy that sat next to me is now a famous actor. He was from Swingers. I'm blanking on his name right now.

Hsu: Vince Vaughn?

Grignon: Vince Vaughn. He's in my yearbook and I did his homework for him. He wasn't, like, an asshole or anything. He was just like a—he wasn't like a bully. He was just like just an absent—he's was taking it because it was a computer class and he was a senior and I was a freshman and—whatever I was at the time—and, yeah, so I did Vince Vaughn's Fortran homework at <laughter> Lake Forest High.

<laughter>

Grignon: Of all things, but, yeah, that was kind of my first formal programming class, where we learned—and that's actually the programming classes where space matters, right? So, you kind of—like, two

spaces means something and no spaces means something and you know. Now, of course, it's a very different world. But.

Hsu: So, you were taking programming classes at Iowa State.

Grignon: University of Iowa.

Hsu: Oh, University of Iowa.

Grignon: Which is Iowa City And so, yeah, I went to school at University of Iowa and I was never a good student. I was actually a really terrible student. It was fun because there was enough—I guess this captures what we would consider kind of the learn-at-your-own-pace today. But at the time you had to follow the curriculum, right? And anybody's who's kind of a self-motivated learner would just go right to the interesting parts in the book. You'd get the curricula for the year or the semester or whatever it was and then you would go right to the thing that looked interesting. You'd learn enough to just be dangerous with that little thing and then, okay, semester's over as far as I was concerned, right? And so, I was a horrible, horrible student. I was always going in to meet with the dean. I was—my grades were just on the—sometimes they'd be Fs. Other times they'd be Ds or Cs. I was not a good student. But it was because I wanted to learn at my own—I wanted to learn what I wanted to learn, which, you know, in retrospect there's kind of what I call "eating your vegetables." You know, now that I've kind of grown up and I talk and I run engineering teams, you know, there's this thing that everybody has to do. Nobody wants to write the utility functions. Nobody wants to maintain code. Nobody wants to bug fix. But you gotta eat your vegetables. It's good for you in a way. And I had never—that lesson of learning to eat my vegetables didn't sink in until much later. But in the early years of computer science I just wanted to do the fun things, right? Which was Assembly programming, Pascal—those were languages that were actually useful on the Mac. Pascal was one of the big language at the time, before C, and those were things that kind of furthered my own personal agenda. Things like data structures, operating system concepts, compiler—all those were classes I took and I didn't do that well, but they were interesting or parts of them were interesting, but it wasn't—I didn't go through that kind of rote mechanical process that they wanted you to learn. So, as a result, I had to ultimately go back later in my career. Because, funny enough, while you're going through it, because it wasn't interesting in the moment, you'd go back and re-visit these things and be like, "Oh! Now that I'm working at Apple it would be really interesting to learn, you know, some of the mechanics of compiler or code generation or code gen." Because, funny enough, compilers produce bugs and you'd have a bug that wasn't necessarily in your code, but in the compiler that produced the code and, so, you'd have to start de-bugging. So, that's where, all of a sudden, all this stuff came full circle. You'd start to learn some of that stuff again. And that was much later on in my career. Not in the educational kind of time frame.

Hsu: Really. Interesting. So, you met a future colleague there.

Grignon: At the University of Iowa, right. In fact, and it was a good friend of mine. We worked at the computer science—there was a multi-media center as part of the university. This guy Jeremy Wyld and Jeremy and you actually worked together. I met Jeremy at Iowa and he was just the hotshot programmer guy and I was trying to learn how to become a hotshot programmer guy. But, yeah, we met and the multi-media center was working at the time with Apple on the Newton. So, the Newton was a new PDA. I mean, the PalmPilot I think was out then. But the Newton was—or was it not?

Weber: Not in—what year was it?

Grignon: This would've been '92-'93.

Weber: No. PalmPilot's '95.

Grignon: '95. So, it was before then, huh? And so, we had—the Newton was a thing that was like this. It had a lid on it, had a stylus, and you'd write on it. It was—but what was interesting was the architecture of the operating system, you know, they had these constant—this concept called "soups". It was the way that data was structured—programs actually executed. It wasn't a fixed thing with a hard drive. It was just this—you could insert a card and, all of a sudden, programs were there. That was not really a concept that we had in active consumer electronics at the time. And so, we started kind of hacking on it and there was a company that made a wireless card. It wasn't Wi-Fi. Because Wi-Fi wasn't thing. This was the early days of wire*less*. So, it was no man's land of data connectivity. And we got this wireless card, which was PCI, I think was the name—PCMCIA was the name. That was the name of the interface. And so, you'd plug in this card and we got this thing to work with the Newton. And so, we had a very early prototype running inside of the university where we had a Newton with wireless connectivity. It wasn't Wi-Fi. But it was still cool to be able to walk around and you had access to the Internet. So, this was the early days. So, Mozilla had just come out, right? So, now we're kind of in the '95—

Hsu: Oh, Mosaic?

Grignon: Mosaic. Correct.

Weber: '93. Yeah.

Grignon: '93. Yeah, I'm really bad with memory.

Weber: Could you run a browser on the Newton?

Grignon: Well, we had a really basic thing. It wasn't a browser. Remember—there was something—Gopher—

<overlapping conversation>

Weber: Telnet, or something.

Grignon: There was Telnet. There was—

Weber: There were line-mode browsers you could've used.

Grignon: Yeah, they were really basic things. And you could do—you just had basic connectivity. And that was it. And so, those were the early days of what could we do with, you know, the Newton, if we had this concept of Internet, of connectivity. We didn't know it was really called the Internet at the time. It was just what if we were able to have connectivity to the hospital system? Iowa hospitals are really good and one of the things that the multi-media center was looking at was how to embrace technology for the health-side of things. And so, that was what we were kind of chartered with. But that turned into what we now know as just screwing around on the Internet. Like, let's just do all the random things aside from work!

<laughter>

Weber: So, what you could do on a Newton at the time—you could do email, clearly, through Telnet or something like that.

Grignon: Yes.

Weber: You had some form of browser.

Grignon: There was. Now, funny enough, a guy I didn't know at the time, but would feature prominently in my life later on wrote one of the first web browsers for the Newton called Newtscape. His name was Greg Simon and Greg and I started WebOS together over at Palm much later on in our lives. But Greg and Jeremy knew each other, but not until Jeremy had been at Apple for a little bit. So, it's this weird, like, you know we all—we kind of all talk. Anybody who's been in the Silicon Valley or been part of this culture growing up, we always say how small the Valley is. And then you start to see this really intricate set of connections that we all have. Like, "Oh, I used your thing before I knew you," or anything—but it's these connections that surface much later on and "Oh, by the way, we started an operating system together,"

and all these other things. Funny how that all kind of plays out. But, yeah, those were—so Jeremy was one of my good friends. And he ended up getting a contractor position with Apple in Newton. And so, it was like this thing. He was dating this girl—now his wife—and we were all good friends and I remember he got this job. So, he was going to be flying out. He'd finished his degree at Iowa and he flew out to Cupertino to go work for Apple as a contractor on Newton. And I went out to go visit him over the summertime—summertime, yeah—

Hsu: This was '90—?

Grignon: This would have been, like, '94. I mean, it sounds sketchy when you think about it today, but he was living with this kid—fifteen-year-old kid. Sixteen year-old? Fifteen year-old. He wasn't legally allowed to work at Apple at the time. His name was Tom Williams. Tom actually invested in my start-up later on in life, but at the time he was fifteen. And him and Jeremy had an apartment in Cupertino. And Tom was a super-smart kid, recruited personally by John Sculley to come to Apple and Tom's mom obviously—he lived in Canada—Tom's mom had to sign off on him doing any of this. And he needed a place to live and I forget how Jeremy and Tom got matched together, but they had an apartment and I remember I came out that summer after my school and I flew out here. No, no! It wasn't summer. I think it was during the fall. It must have been in the—it was raining, so it must have been January. It must have been January-ish. I flew—so, it must have been Christmas break. So, I flew out and I crashed on their couch for a few days, but what was interesting about that and very fortuitous for me was Tom was working as part of the QuickTime team. And Tom and I just kind of hit it off. He's still—he was a kid—*I* was a kid. I mean I was twenty whatever it was. Twenty-one? Twenty—I don't know. Whatever it was then. And Tom introduced me to a bunch of people on the QuickTime team. That's where I met Peter Hoddie. I met what would ultimately become my first manager, Eric—I forget Eric's last name. Eric was part of ATG, the Advanced Technology Group at Apple and Eric Hof—

Hsu: Hoffer? [Hoffert is the correct name]

Grignon: Hoffer? Whatever. Eric was—he was in ATG, the Advanced Technology Group, and Eric was working on this brand-new technology part loosely associated with QuickTime, called QuickTime Conferencing. And he introduced me to them and I went away with a CD that they were giving out, that they were about to give out, at the developer conference in June. It was an early preview of the QuickTime Conferencing SDK. And I left with that CD and we started talking about different ways we could use video conferencing and I got back to Iowa and I loaded it up on our Macs at the multimedia center. And I started programming with it. I was like, "Oh my God we could do some cool stuff." So, I talked to Tom and we ended up writing a concept app, which was based on *Star Trek*—it was *Star Trek*-themed, but it was a video conferencing application based on *Star Trek*. What was interesting about that was at the time the video conferencing apps were Mac OS, you know, 7-styled, right? They were boxy windows. You know, the SDK had the ability to embed video into other things. But mostly it was intended to be used as a window. So, we uncovered a bunch of bugs during the course of that, because, hey, you

know, things happen. And we wrote this app and it was a very novel application for QuickTime conferencing. I mean, it would still have the same—you could initiate calls, you could answer calls, but they were all themed windows, and they liked it enough that they asked if I would present it at WWDC. And so, which was Apple's developer conference at the time. It was down in San Jose. Funny enough, it went to San Francisco for a while. And now it's back to the San Jose Convention Center. And so, I gave this, this talk. So, but I had—I mean, I didn't have any video conferencing cameras. The webcams weren't a thing at the time. But we had cameras as part of our media work at the media center. And so, I did this video conference, a live thing to, I think, a thousand or two developers. But I was at—when you're on the camera you don't really see who's on the other side, so, it was fine. But I was in Iowa and they were obviously in San Jose. And I remember before I did my thing I bought some corn, because everyone just associates Iowa with corn. And so, I'm doing this thing, I'm talking about this app that we built and I had my corn kind of out of, out of frame. And as I'm sitting there talking just looking still at the camera, right? I march this ear of corn across. For whatever reason, people in California thought corn was hysterical and that's where my now email handle "Cornboy" came from. So, everyone just started calling me Cornboy. And at the end of that, they asked, at the end of the developer conference, they asked if I would be interested in joining QuickTime as an intern. And that's when I ended up at the Advanced Technology Group working for Eric Hoff? <laughter>. Hoffiter? <laughs> God, I hope he doesn't see this, because that would be really embarrassing. And I ended up working for him and probably the most fun I ever had in a summer internship. I hadn't failed out of college yet. I was really close, but for whatever reason I bypassed, you know, the loop and I got right in to working for what I think is probably one of the more prestigious parts of Apple, the Advanced Technology Group. I mean, I was from a no-name school from a computer science perspective. It wasn't Stanford, it wasn't Berkeley, it wasn't any of the cool schools. And I was a sub-par, below, well below average student and, all of a sudden, I'm right in the middle of Apple, right? And that was a really impactful time. So, I decided—I remember going out there to Cupertino that summer for my internship and I was like, "I'm gonna make the most of this, because I lucked into this thing somehow." And I worked every day. It wasn't—but to me it wasn't work. It was just what else am I gonna do, right? I mean, if I wasn't working at QuickTime Conferencing I was gonna be writing apps for the Mac or whatever. And so, I just lived there and Apple had a library at the time. Apple's library had this huge patent portfolio and, so, you could just go to the library twenty-four hours a day. This was all new to me. You could just sit there and browse, whatever, all the magazines, all of the patents, all of the books. I mean, it was a nerd, like, heaven. And I did. And so, they had a fitness center, but you don't get a body like this by going to fitness centers. But I went there every now and then. And it was just a really, really awesome time. And, you know, Apple obviously treats their interns very well. So, at the end of it, I just didn't want it to end. But I had been in the middle—I did my little presentation on QuickTime Conferencing. You know, there was a little wrap-up, you know, program for interns and at the end, they were like, "Hey, you know, we have some features in mind for QuickTime Conferencing. Some of the stuff that you've been working on, we'd actually be interested in putting as part of the SDK. Any interest in continuing your internship from Iowa?" I'm like, "Hell, yeah!" <laughs> All the interest, right? And so, they schlep me back to Iowa with about ten thousand dollars in Macs. It was two really nice 840AVs, I think, or 8100—the 840AV was the old one. 8100, it was a PowerPC. They sent me back to Iowa with two monitors, like the big ones. Like, the twenty-one-inch monitors, the CRTs—the stupid heavy ones, right? I had two machines, because you can't do video conferencing alone. So, I had two of them. Two monitors. Two of all of—and they hooked me up with a—the early Internet at the time was IDS,

ISDN. It was before any of the DSLs were around. So, ISDN was akin to frame relay, it was very early on. Iowa had some of it. It was stupid expensive, but they hooked me up with all this stuff. And I was like, “Oh my God, this is great!” So, guess what happened? I went to school, fall semester: You have your choice between writing your tenth calculator for Comp Sci whatever or working on cutting edge video conferencing stuff. What wins? <laughs> Okay, cool: All the video conferencing stuff. And so, my grades just went to the crapper. I had probably the worst academic performance that fall semester because it was just flooding my brain with all of the fun stuff, going back to those early days in college where I would only learn the things I wanted to. This was just, like, everything on steroids, right? And so, I just, I worked on programming all the time. And I loved it. And, at the end, my grades were awful. I mean, if I wasn’t close to failing out I was—I may have been. I don’t remember my grades, but there were a lot of Fs. And I remember I had a conversation with my—Hoffer? <laughs> Like, this is gonna kill me not remembering his name. And I had a conversation with my manager and, actually at that point, I moved managers to this guy named Ramiro Calvo [ph?]. Ramiro and I are still friends to this day. And Ramiro had this conversation. He’s like, you know, “Any interest in joining us full-time as an employee?” And I’m like, “Well, I don’t have my degree.” He was like, “That’s not a problem. We don’t condone it, but it’s not a problem.” And I was like, “Oh. Grades aren’t looking so good. Offer from Apple.” It wasn’t ATG; they were actually spinning this into part of QuickTime proper at the time. I was like, “Yeah.” So, I think I had the most awkward conversation with my parents. “Hey, thanks for spending all the money on out-of-state tuition to a state school”—’cause I was from Illinois—”thanks for all that, but I’m gonna just stop. And I’m gonna go get a job.” And they were like, “Uh, no. You’re gonna keep going to school.” And I was like, “Uh, no. I’m gonna go to Apple.” And we ultimately, obviously, settled on I would just go. And I didn’t have a formal offer in hand. I was just a verbal kind of a thing, and I got on a plane. So, I didn’t enroll in classes for that spring semester. I got on a plane. Southwest Airlines at the time—they had seven layovers to get from Chicago Midway to San Jose. It was like the Greyhound of the sky. It was like Southwest tour. And I landed somewhere during that, you know, seven-hour flight—seven [hour] layover, whatever it was. In air, one of my roommates that I was gonna be living with—we had this whole Iowa mafia is kind of what it was and we all; Jeremy started it. And then a few of us kind of packed on. And so, I had roommates set up, all from Iowa. We all knew each other. And I landed in San Jose, I still didn’t have or knew what my prospects of this job, aside from a verbal, “Yeah, let’s do this.” We went out to lunch and they’re like, “Hey, while you were in the air I got a call from your manager Ramiro and they’re offering you a job. The pay is 52,000. We said you’d take it.” <laughs> It was like, yeah, that’s about right. And that was when I learned that I had actually gotten a formal Apple employment offer. And so, I went over later that afternoon and I picked up the paperwork and I signed it. I didn’t even read it. I was like, “I’m in.”

<laughter>

Grignon: And then I started. And that’s how my career at Apple started.

<laughter>

Hsu: So, how long were you, were you in that group, QuickTime Conferencing?

Grignon: So, QuickTime Conferencing became an actual product. And more than just a piece of software we built a camera that had composite video that plugged into your AV-enabled Mac. And that lasted for a couple of years. We had layoffs though. So, this was the hard times of Apple. And this was—this must've been, like, '90, '96-ish? '97? When did we have our big layoffs? The layoffs were interesting because, okay, now I'm an employee, right? And I was one of the first on the chopping block. Youngest, most recent, whatever: first. And I had gotten laid off and, so, I forget how long this was. This was—must have been '97, something around there. I had gotten laid off and I was just wrecked. I was like, "Man..." But across the street from Infinite Loop was Sun at the time. That ultimately became the building where iPod looped into. Mariani One. But Sun had their campus there, their Java campus. And so, they got wind of layoffs at Apple. They had, they flew a plane around Infinite Loop—one of the big biplanes with one of those big banners on the back—and it's like "Sun's hiring," on layoff day as these people are getting news that they're getting laid off. And so, I went over. I went over and I talked to the multimedia team at Sun and they were like, "Let's do this. This is a perfect skill match for what we're trying to have at Java. This is great. Come and join Sun." And I was ready to sign on the line and I got called back into my manager, Ramiro's office, and he's like, "So, about that layoff, we actually are gonna do something different with the team and we want you to be part of that." I was like, "Well, but I'm gonna—I'm about to go sign with Sun." He's like, "Don't go sign with Sun. We want to keep you." And that was good for like a forty-thousand-dollar pay bump. <laughter> I got this huge, huge pay raise out of this, like—and that was actually an important lesson. And it became a thing that, you know, to this day is still true: The only way to make money at Apple is to quit and come back or to effectively threaten to quit and come back. And so,—and so—I stayed. So, they took back the layoff, I got a big pay raise, and then we started what became QuickTime Streaming. And so, QuickTime Streaming was the early days of streaming. Real Networks, you know, it was a thing at that time. And we wanted to do the same thing with QuickTime. And so, we were gonna kill QuickTime Conferencing, the video conferencing aspect of it, and turn that knowledge base that we had developed into a streaming platform. And, again, if you think of that, it's just, "Well, duh." But at the time you had two five-hundred-megabyte movies, whatever, like, these gigantic files that had to be resident in order to be played. And so, Real comes out with this, oh, they can live in the cloud, but we didn't really have the cloud then.It was just on a server some place. So, what if we were to do that with QuickTime? And so, that's the birth of QuickTime Streaming, which is now just folded in. And so, we worked with Real. So, this was all standards based. I helped define a protocol at that point called RTSP. RTSP is, you know, really nerdy low-level thing, but it's how a streaming session is set up. And so, I got my name listed on the, on the thing, the working group, whatever the standard—the, I forget the name of it then. But the RTSP Protocol; so I'm on that and I got to work—we actually flew up to Seattle to go meet with Real. And it was a really fun experience. And we built a whole server on the backend in Java. So, I got to still continue the Java love. And we made a whole thing around video streaming. At that point though, I was done. And it was time to move on, right? So, you kind of feel like things are happening, I'm done now with RealTime video. This was in '98-'99 and Jeremy, my buddy from Iowa, had moved on. He was still at Apple this whole time. They had obviously disbanded Newton. And he went and joined this company called @Home Network. And @Home had this crazy idea of using cable, your cable TV, to get Internet over. And better yet, they could give you crazy-fast speeds over your cable modem. Like, whoa,

that's cool. So, Jeremy was over there working on some of their apps for the Mac and he's like, "Hey, you want to come on over here? It's kind of cool and fun and we're doing some neat things. We actually want to build a set-top box." And I was in. I was like, "Yeah, let's do that." So, I interviewed, got the job offer, and, so, I moved over to @Home. And we were building a set-top box from scratch, right? It was Windows CE-based. It was a—so, obviously, AT&T, I think, was our cable partner. TV Guide was building the, the—remember when *TV Guide* was like a little paper book? Well, they were building the software for the EP. the program guide, and we were supposed to make the overall look and feel. And so, it had some element of realtime video. So, that was kind of a nice fit, but we needed to create this framework on Windows. So, it was my first foray into programming on Windows. And we started building the set-top box. And the cool thing was we had: there's this thing called a head-end, which is, it's effectively a cable subsystem, right? It's where you actually get your cable TV from. And we had one of those inside of @Home. And there was one RF guy that was ostensibly in charge of it, but it was really my head-end and we got to decide what channels went where. Like, what content—like, Channel 2 is what? It doesn't have to be Channel 2. It can be, you know, CNN, it can be whatever. And, of course, we had adult channels, we, <laughs> and we mapped them. We did all the terrible things that you shouldn't do. But we did that, right? 'Cause it was @Home. It was the Dotcom, you know, heyday. And it was a lot of fun. We built this really cool set-top box. And then, you know, the bubble popped. And when I joined @Home, my stock—my options were priced at eighteen dollars-ish and these were the days where Webvan, Pets.com, all these crazy things and it was just company of the week. Like, it was, money was everywhere. And I rode our stock all the way up to two hundred and change. Two hundred and two, something like that? And it's all single share. And, then, the bubble popped. And it started to go from two hundred and two down to a hundred and eighty, a hundred and seventy. Well, if it was a hundred and eighty last week, I mean, this is a steal. I'm gonna keep holding on. I rode it all the way into the ground. And so, a single share—Jeremy got five hundred shares as a referral bonus, an employee referral bonus, at the time; and at the height, when he finally sold it, I was worth fifty thousand to him. <laughs> It was just stupid dotcom Internet money. And him and his wife took me out to dinner. So, that was kind of a nice thing. But that was the only real money I saw out of Excite—we ultimately, I'm sorry, @**Home**. @Home during that time merged with Excite, so it became Excite@Home and then rode it all the way up and then all the way down. It was like that movie *Dr. Strangelove*: Slim Pickens is hopping on that missile and he's just like, throwing his hat, "Whoo!" as he's riding his nuclear bomb down. I kind of felt like that as we were watching, you know, the Excite@Home crater. We had at that time—we had senior software engineers disassembling cubes. <laughs> It was kind of an incentive to move on. And people kind of hung it out, right? But I got the hint. And it was time to go. And so, I joined a start-up called Pixo. And Pixo was started by a bunch of ex-Apple people, and their idea was creating a mobile operating system for mobile, just devices. It wasn't phones necessarily. It was just anything mobile. And so, two-way pagers were still a thing. We had a deal with a company called Scout. And Scout had made, using the pager network, kind of an early two-way communicator. And we were building their software for them. That was, like, our, Pixo's first big success. And so, I joined on. We were still building on Windows at the—Windows was our build chain, even though we were all Apple people. And so, we built—we'd gotten some really interesting deals. We had some success with Nokia. We built—we cloned the Nokia operating system with our stuff. You couldn't tell—

Weber: Symbian or—?

Grignon: What's that?

Weber: Symbian?

Grignon: Well, Symbian was their operating system at the time and the phone was, I think, the 6100. It was—that was the hardware that they had been—and that's what we had developed and we deployed; we showed it. It all worked, it was great. But it didn't really make Nokia look good. And so, they killed the project even though they were big fans of, of our little start-up. But around that same time Tony Fadell was at Apple and Tony reached out from this Apple network that we had and he was basically like, "Can you make your stuff work with this?" And it was one of those jog wheels. Those video editing circular wheels, that was a USB accessory. And we're like, "Oh, yeah." So, a bunch of people made that happen. We had already made the operating system color, but he wanted it in black and white and that wasn't a problem. And as a—kind of after we demonstrated that he was like, "Well, can you make it," like as an example, "play—just give us an app that played MP3s." And so, I had already written, in the early days when I first joined Pixo, I'd written an MP3 player. Just as like a test, like a fun thing. So, we revived some of that code and we wrote this app and programming the source code has, you know, names, you know, that—and our name for this app and the source code was—the class name was TMP3SampleApp, as just an example. And, obviously, this was the iPod and we didn't know that we were working on that. And, you know, we had hardware show up in boxes, that was totally like big wooden things, and you could use it, but, you know, we weren't allowed to see any of this. And the next thing I know Steve Jobs is onstage introducing the iPod, you know, that iconic picture of this thing. And I was like, "Oh, man, we did that! How c—I mean, this is awesome!" To this day every iPod ever sold, the app that gets launched is TMP3SampleApp. They never changed the class name and the source code. <laughter> It's still like—obviously they're not making iPods any more, but that was—that stemmed from those Pixo days. And when I saw that, you know, the first iPod wasn't really a success by any stretch, right? It was a thing, but because it was tied to the Mac, you know I was like, "Mmm, Apple, they're doing some cool stuff. I kind of want to go back." Jeremy, again, from Iowa, joins Apple in the Cocoa Team. <Hsu assents> And he was doing frameworks of some kind. And I decided, you know, "Pixo is now the worst deal ever!" And best deal that Tony ever wrote I think. He got a license for Pixo's source code, it was like $50,000 or something, in perpetuity. No per unit royalties. <laughs> I mean, it was like the best deal ever for Apple. And we had—worst deal ever for a startup. And we were on our way to cratering, and, so, I decided to join Apple. And so—

Hsu: Pixar was on the way down.

Grignon: Pixo, yeah.

Hsu: Pixo, I'm sorry, Pixo, yeah.

Grignon: Pixo was on its way. You know, we tried to reinvent ourselves with some server technology, And some other things, but that wasn't going anywhere.

Weber: Could you talk a little bit more about what Pixo was before the iPod, I mean, what was the intention of—

Grignon: The intent was an operating system for mobile devices. Meaning, we could live on top of an RTOS, you know, Real Time Operating System. We could live on top of the Mac—or, you know, as a simulator. Or Windows. Or we could just be the entire thing ourselves. And so, you just had a chip. We had a little hardware abstraction layer. You had ported to that particular thing, and then we would be up and running. So, the whole point was hardware with very minimal resources. Today's version would be, you know, like a Raspberry Pi, but even that is infinitely stronger, hardware-wise, than what we had at the time. But imagine Raspberry Pi without Linux, and you just have a chip, and stuff. And you need to make it do things. You have a screen and a chip, and some memory. What do I put on there? And so, Pixo was this platform that had a menuing system. It could accept input. It could do things. You know, if there was infrastructure there to play music, for example, we would use that. If not, you could write some drivers and interface with us directly and then we could tell, you know, the audio chip to start playing this, you know, audio stream. It was those kinds of things.

Weber: And the intended audie—I mean, Nokia is one, but this mobile phone—

Grignon: It was any kind of mobile device. So, we were kind of shooting in the dark. At the time, Scout was our big—you know, but again, this was pagers with up and down arrow buttons on the side. You know, four buttons on the bottom, but it had a nice big screen with four lines and graphics and things like that. So, we could work on something as little as that, or up to a phone. And so, for the Nokia device, we replaced—we made it pixel perfect. You could even play that iconic *Snake* game, remember that, on Symbian, you know, with the arrow keys. We replicated it down to the pixel. And so, that was the intended audience, was, as an operating system for mobile class devices. And a very natural fit for the iPod when you kind of look at it from that perspective. But yeah, we weren't successful. I mean, just again, another example of, startups don't necessarily turn out to be Instagrams or WhatsApp, or whatever. It may be technologically a great thing. I mean, you could argue this was probably one of the more successful startups of all time! We built the software that powered iPod! <laughs> But we never saw commercial success. And, for whatever reason, you know, that, that went that way, but it was time to go back to Apple. And so, that was my—that start, that started my second stint. Apple was looking at creating an app called iChat, at the time. iChat was going to initially just be an AOL Instant Messenger-based messaging <clears throat> thing for the platform—app. But they had dreams of making it into a video conferencing product. And so, that was the interest in hiring me. And so, I came back to do the video conferencing part. But we still had to write the actual chat app. And so, we did! So, we wrote iChat. So, there was three of us at the time. We packed the team, I think, up to five or so. Still friends with them till this day. And we created iChat. And we introduced it, and that's where I started to, started to work with Steve Jobs a lot

more. And iChat was kind of the new sexy app in Mac OS back when Mac OS features were still being added on. Kind of, you know, we didn't have the calendar. We didn't have iCal. We didn't have, you know, certainly not iMessage, or whatever they call it now. It was going to be iChat. We didn't have any of those apps, and so, we were packing them on. And Steve had been starting to pay a lot of attention to these new apps. The look and feel. The operating system itself was undergoing a pretty drastic change visually. You know, there's the introduction of Aqua, which was the look and feel of the bubbles. The, you know, the green and red and yellow buttons in the windows. And so, Steve had—I'm sorry—Scott Forstall had a weekly meeting with Steve, where it was in a room called Diplomacy, IL2, Second Floor, which is where we were. And we would show Steve—well, Forstall and his crew, Greg Christie and everybody else, would show Steve new designs for whatever. And that's where I started showing up. Forstall was really good at putting other people in front of Steve. So, it wasn't just him. It wasn't a power grab; he'd already, you know, become Steve's, you know, favorite at this point. But he was really good at getting other people in front of Steve. And so, I was one of those people. And so, I started showing him, not just iChat stuff. We were looking at bubbles and just the early des—when it was just text. But when we started to introduce the video, you know, the video feature, it was the concepts around that. And, in addition to iChat AV, we realized we needed to make a camera, in parallel, because people didn't have—webcams weren't a thing. We weren't built into the Apple, you know, product mix at that point. And so, again, Tony Fadell, the guy that brought the iPod to life, was in charge of getting this camera—it's called the iSight—bringing the iSight to market. And, in software, we had no expertise in dealing with hardware. But I did,right? From all that time in working with Nokia, and you know, hardware wasn't a thing to me. So, I was the guy—since I was leading the development side of things for iChat AV, I seemed to be natural fit for working with Tony and his, his merry band on bringing iSight. And so, I worked with them, I went to their staff meetings, and I was like the software mole from Forstall's organization into Fadell's organization. And—but I really enjoyed that kind of thing. I mean, working on hardware is so much fun. And I really enjoyed the challenge of building, like, this camera. We had some really crazy ideas of stuff that we wanted to introduce with this iSight camera, like auto-focusing, you know, based on kind of hotspots in the screen and things. And so, they actually listed me on a patent or two from them. And I got to just be good friends with Fadell. He was a software guy, he needed some software structure to his hardware organization. And, you know, he obviously kind of remembered me from the Pixo days, and we just hit it off! And so, we started that relationship during the iChat/iSight AV—iChat AV/iSight days. And then I was still part of the Steve meetings and Tony was brought into the Steve meetings every now and then. And so, that's kind of where all that happened. That's also the—that's also where I got my nickname. I know that was <laughs> something that I'm know—that I'm known for. In fact, it started during the iChat days. And we had been showing Steve very early builds of iChat. And so, this one particular day, we're, we're waiting in Diplomacy, that conference room. And Steve's admin would give us a head's up, a call when he was on his way. And she would usually give us kind of an attitude report. Like, "He's in a good mood," "He's in a bad mood," "Just had his teeth worked on," or whatever. And she's like, "Yeah, he's in kind of a weird mood today." And so, we're like, "Okay, whatever." We're just hanging out. There's like six or eight of us in this room. And <clears throat> he comes in, he plunks down into this, this chair that he always sat in, and he's just kind of shaking his head. And he says, "Weirdest thing happened. You know, I'm going through his email," he went—he's saying, "I went through all my email, And somebody called me a 'fuckchop.' It was a customer who was really pissed, about whatever, and they called me a fuckchop." And we're just like laughing, like, "What?!" He's like, "What the fuck's a

fuckchop?" And we're like laughing about it, and you know, and everyone's like, "Whatever. It's random." But Steve—for—to his credit, he read every single piece of email that he received. Didn't necessarily respond. You know, some of the more important ones he would fire off to somebody. Every now and then he'd respond. But this particular one, it just kind of sat with him. And he was like, "What the—is a fuckchop?" Anyway, so we launch into our demo for the day, and I'm up. And so, when you show video-conferencing, obviously there's the machine in the room, but you have to go to another place to demo, to do the demo. So, I set everything up. You know, we kind of talk him through some of the design decisions that we'd made and the app, and we—whatever. Time for the software demo. So, I run off to the other room. It was somebody's office. And I have one of the guys who's working on the actual video-conferencing part, because it's got bugs, right? So, it worked-ish. <laughs> And we start our video chat. And fine, it's going well for like the first five or ten seconds or so. And, eventually, the audio craps out. They can't hear me. They can see me. I can hear and see them, on my side. And if there's one thing that I learned from all of the iChat demos, all of the meetings that I'd been in, is Steve is very impatient—was very impatient. And the last thing you ever did was sit there and try to fix problems. Like, just cut. Just stop. <laughs> And move on. You'll get like a little verbal lashing, and that's it. And we're going through this demo and this weird situation happened with the audio crapping out. And it's not something that we had seen a lot of. So, we're trying to debug it. And the guy who was working on it was with me and we're on the terminal, and we're going through, like, looking at all the debug spew to try to figure out, like, "How is this happening?" And while he and I are kind of looking at things, I can see kind of out of the corner of my eye, the screen's still up and running, and there's awkward silence. You know, there's Steve getting impatient, and then he turns over and he starts grilling them—whoever's in the room still—about whatever. This icon, this look. Whatev—some whatever random thing. And I can see him just getting more and more agitated. <clears throat> But I'm tone deaf to these cues. And he starts like being more bursty [sic] and more angry with the people in the room. And—but, again, I'm ignoring all this, or, or whatever it is. And he finally stops, and he's like, he said, and he's looking right at the camera, and he says, "You know, I finally figured out what a fuckchop is! It's you! Stop wasting my goddam time! We're done!" And I just see this frame of him looking up into the camera while he's got the mouse looking for the close box on the video chat. And that's the last thing that we see, and I'm just like, "Oh <laughs> okay." It was just like that sinking, just like, "Ugh. Hmm." And, after that, the people who were still in the room, they just started calling me Fuckchop. And it was like, "Hey, Fuckchop." <laughter> And I'm like, "Thanks." <laughs> Right? And it just became my nickname. And it wasn't a good nickname, because, you know, obviously, I just failed <laughs> on this demo. Steve was super pissed. I didn't go to any UI reviews for a while after that until we actually had more things to say. He never really held a grudge, but it was just like 'a thing,' and it didn't feel good. And—but people kept calling me Fuckchop. And I was like—and it stung for a long time. Well, eventually, I wrote a feature for the Mac, called Dashboard, which would feature later on in life. Dashboard was a way that we could use this up-and-coming technology, WebKit, that we had, which was the browser, Safari. How could we use WebKit to do these little miniature applications? So, web pages were the thing that we knew of. They were big boxy things that sat in the frame. Back to those—that early day of QuickTime conferencing, we don't have to put things in boxes, alright? We can put them in *Star Trek*-themed things. And, in our case, we were putting them in nice little alphaed out rounded windows, right? There was no frame around them. Widgets were these tiny little lightweight things that could animate <makes sound illustrating animation> up and down. They were super cool-looking. They could be perfect circles like a clock, and it would bleed through. You would never know it

was in an actual square under the covers. And, anyway, so I built that. And we'll come back to that, I guess, in a second. But, so, after Dashboard, I was kind of—eh, I didn't really—I didn't know what I wanted to do. I was kind of bored in Forstall's org. Mac OS was starting—the feature set was starting to taper down. And I've been talking to Tony from the iSight. We'd still maintained our friendship. And he's like, "Come on over. I got a couple of really cool things that you might be a good fit for." And so, I joined Fadell's organization. But it was a weird reporting structure. So, I was working for two people, Steve Sakoman and Jon Rubinstein. And Sakoman had gained, you know, his notoriety from starting BeOS. Co-founded it with Jean-Pierre, I think?

Hsu: Jean-Louis.

Grignon: Jean-Louis Gassée.

Weber: Gassée.

Grignon: And—

Weber: But also Newton stuff early on.

Grignon: Newton days—Newton things from early on. So, things are kind of coming full circle, so I worked with Sakoman and Jon Rubinstein who is running hardware now. Jon had stepped down from his—all of hardware—role, into just focusing on iPod. And, to loop it back on the fuckchop story, so I joined, you know, this organization. And, you know, a year has gone by, and I thought, you know, I'd come to peace with this nickname. And so, the way it works at Apple, when you want new business cards—I'm now in a different building, I'm in Mariani One—the way it works at Apple is you go to the Apple Printing webpage, and you get your business card. And so, I put 'Fuckchop' as my title. <snickering in background> You can write anything you wanted. And I did F, star, star, K, Chop. [F**kchop] That's how I wrote it. And I submitted it; it was like a Friday, you know, whatever, Friday afternoon, submitted my card. And, like, two weeks go by, and I don't get my cards. Usually Apple Printing was on top of things. So, I go down to Rubinstein's office, and kind of rap on his door. And I'm like, "Hey, Ruby, did you see my request for my business cards? My, my thing?" And Jon had these, like, like, these glasses, and he's, like, working on his computer, and he kind of turns at me, and glasses on the rim of his nose, right? And he's like, "I've been meaning to talk to you about that. What the fuck is wrong with you!? You can't put that on Apple stationery. <laughs> No, I got it. I was there, I heard the thing. But no absolutely not!" I was like, "Alright, fine." So, go off to lunch with my friends, and we're laughing about it, you know, because now again, I've adopted this name. And we're thinking of other random titles. And I come back, and it's, like, two or three in the afternoon, and I get this email from Apple Printing saying, "Your business cards have been approved!" <laughter> And I'm like, "What the hell is going on here?" Right? And I decided to just, you know, button my lip. And sure enough like three days later, in Campus Mail, come this box of

cards. And it's, you know, "F, star, star, K, Chop." I was like, "Ah, this is great!" Clearly, Jon had a change of heart. As much as he was just, you know, blowing at me. So, I get a card, and I go to give it to him. I go rap on his door. And he goes, "I just thought you might like one of these cards. They're really funny." And he's, like, "Don't you dare show that to another person. This is contraband!" <laughter> I was like, "What's going on?" Right? Turns out he got the same email I did. It turns out, it was two weeks to the hour that I had submitted my request. He's an EVP. He's got other stuff to do, right? So, he was in a full day of meetings, decides to push off to Monday. Calls on Monday to get the card canceled, because he didn't want this on Apple stationery. And they were like, "It's too late. It's already been produced. It's on the way." <laughs> And he was like, "Mmm." So, it turns out, two weeks to the hour, unless your manager explicitly approves or rejects your card, it gets auto-approved. <laughs> They've changed their policy since. But <laughs>—and so, this card that I've got is this testament to like a broken process, but they can't do that. You don't get away with that anymore. <laughter> Anyway, that's the fuckchop story. And it's—I still have a bunch of the cards. Like, I only have one box, obviously, I couldn't get a reorder, but <laughs> I give them away for random things.

Weber: To circle back a little bit with—so Steve Jobs had come—when you left Apple the first time, Steve had just come back.

Grignon: He had just come back. So, <clears throat> he had acquired Apple, right? This was the whole NeXT thing. That was the joke, which was, "Steve's coming back." In fact, I had a very—when Gil Amelio was still running the show, we had this weird kind of a tipping point causing me to leave Apple, and—to Excite—was Amelio had this big all-hands [meeting] over at the De Anza College, and we were all given these really crappy painter's caps that said, "I was there when the comeback began." <laughs> And it was the worst rah-rah you've ever seen. And it was depressing, right? But what he hadn't announced yet was the acquisition of NeXT or any of that. And that happened pretty shortly thereafter. And when Steve came back, you know, there was this idea that he had this thing called Carbon, which was going to be a reinvention of Mac OS on top of NeXT, and a whole weird, you know, set of things, and we have this huge effort in Tony Fadell's building, in Mariani One, to port Mac OS from PowerPC to Intel. And it was super hush-hush. Like, nobody was ever supposed to know. And those efforts obviously combined—

Hsu: And that was OS X to Intel?

Grignon: That was what became OS X. And I forget the code name for that.

Hsu: Marklar.

Grignon: Marklar! Yeah, good memory! <laughs> And Marklar became this combination of Carbon, Mac OS, and Intel on this random—on PCs. And that's around when I left, back that first time. And then, when I joined back, obviously things had settled down, you know, Mac OS [X] 10.1, I think, was the operating

system we were at, at that point. 10.0 was out, 10.1 was coming. And that's when these new features started to be piling on for iChat.

Weber: Right, so you left—I mean, initially, it didn't give you any great confidence that they were <Grignon interjects>. Oh, yeah.

Grignon: You know I was a fan. In Iowa, at University of Iowa, we were one of the first universities to adopt NeXT—and NeXT OS. And we used to make fun of Steve Jobs. We had—he was very young at the time, obviously, and we had this lab-full of NeXTcubes. They weren't the 'Slab,' the 'Slab' wasn't out yet. It was these NeXTcubes with the black and white monitors, the black—the whole thing was murdered out right? Black mice, black monitor, black Cube, black printers. Like, and it was super cool and expensive. And when the printer would run out of paper, like, all of the machines in the lab would say, <changes register of voice> "Your printer is out of paper." <laughs> Like, and it was like twenty machines saying this. You're like, "Really?" <laughs> But when you set up your first NeXT account on NeXT OS, you got a email, right, in their mail app. And it was an audio attachment from Steve. So like, <changes register of voice> "Hi! This is Steve Jobs. Thanks for buying a Mac," or, "Thanks for buying a NeXT," you know, whatever. And we would impersonate, like make fun of him. It was, like, <changes register of voice> "Oh, hi! This is Steve Jobs." And, all of a sudden, that was from the University of Iowa. All of a sudden, like, we're now owning this operating system that we were making fun of! And—but NeXT always kind of had a soft spot in my heart, right? Mathematica was the app that I was running on NeXT at the time. There was the big science-y thing. And so, I always kind of liked the NeXT. And then when we started to kind of put that Mac OS look and feel on top of this other thing, I hadn't really understood—back to those UNIX, those System V days, it was rusty, but also there's this terminal. It's like, "Wow, are these things, like, [running] at the same time," or is this just an app that is like DOS? Well, no, it wasn't. It was a window into the underpinnings of this, you know, this operating system. So, all those—like this whole, like this career from childhood to getting my IIsi with the FPU with A, with A/UX. All of a sudden, all of sudden, you know, here I am in 2001 programming on this thing. And it's like, "Wow! Like, how often does, you know, your career just come full, you know, full-circle like that?" And, yeah, that's how, that's how all that started.

Hsu: Did you have any interaction with Steve the first time—before you left Apple?

Grignon: No, I didn't.

Hsu: Okay.

Grignon: Hmm. No, aside from going to the random campus events, he was Steve. But you know what, he also—maybe it was because I was on my way out, but he didn't have that same mystique then that he had when he passed. You know, he wasn't Ste—I mean, there were certainly people politically motivated,

who were trying to get in his inner graces, you know just for their careers and things like that, but he was just our CEO, right? Like, Gil was our CEO. Michael Spindler before him. Spindler would give these all-hands, you know? And he wasn't a guy to be revered. None of these guys were to be revered, right? Spindler would get up in front of everybody in these—he always wore like these light blue oxford shirts, and he sweat profusely. And he would have his arms up, and they were just these giant dark blue patches under each one of his armpits. And it was like—but again, these weren't the days where CEO—Apple, at the time, wasn't a company that revered its CEO. They were just leading you at the time. And you made fun of them, you did this. Apple was the company where people would barely show up clothed. People would be, you know, the jugglers would be spinning their pin—you know, those bowling pins in the lobby at atriums, like their little jug—it was just a—it was a circus. And Steve turned the culture around when he came onboard back to, "This is a business. This is my company." And you saw that in a very pronounced way. You know, there were nice little things, like nice little upgrades. And then there were some downgrades, right? So, I remember—back to this body—I would go and buy a donut in the morning at the Apple Café, Café Macs. And Steve decided donuts weren't good for you, and he killed all the donuts. And he killed the Coke machine. You had to use this whatever thing—this other brand that he had put in. And that was good. Again, it wasn't like, "Well, Steve said, it's got to be great." It was just like, "Fuck, man! I want my Coke. I want a donut. I want all the things." And he just—and more than once I got bit by buying a cookie at the cafeteria line, and finding out the cookie was vegan, and it just tasted like crap! <laughs> I was like, "Really?" You know, but those were all like, that was Steve kind of putting his mark on the company. And <laughs> so things like that happened, right? So, he wasn't always, you know, he was Steve in the early days. And then as we started to have some success, and he became this very visible leader in front of everybody, you know, when you look back at, "When did iPod become successful?" Well, Steve saw that we needed to branch out of the Mac. And iPod became successful when we made drivers and iTunes for Windows, and we had that ecosystem approach. All of a sudden, we weren't just limited to our five percent of whatever it was market share, at the time, it was the rest of the market. And that's when people started to see Apple as a viable consumer electronics player.

Weber: But that was over Steve's objections.

Grignon: That was over Steve's objections. But he saw—he could be convinced. Just like the phone was—Steve didn't want to do a phone for the longest time. And he had to be convinced. So, guys like Phil Schiller, Tony Fadell, Rubinstein, had to lobby Steve in a big way to start the phone project. And that's where, one, I was brought in under Fadell, under Rubinstein, under Sakoman. It was because they had this idea of making a phone. And, well, I'd worked on the phone as part of Pixo. <laughs> You know, I built a phone, right? And how hard can it be? Once we finally got the green light to start what would become the phone, we didn't know what it was we were building. Tony had this vision of making a iPod phone. And we built it! Actually, we built several iterations of it. So, me and I think three of us, I think got together, four of us, Jeremy was one of them. And the infamous Jeremy comes around <laughs> back into it all. And we started the phone project with that idea that it was just an iPod that made phone calls. In fact, I've got an early patent—or patent application—I don't think the patent was ever granted—where the input modality was that circular wheel on the iPod. And it was a double-shot plastic that went from blue controls, that was like play-pause-next-previous kind of a thing; when you went into the software, the

OS—Pixo <laughs>, our thing, when you put it into phone mode, it changed to orange, and it was zero through nine, A/B/C, D/E/F in like an old rotary phone look and feel. And you could click in place, or you could kind of move around and the thing would—and you texted and you made phone calls through that interface. And it was as awful to use as it sounds. <laughs> It was really bad!

Hsu: And that prototype used your original Pixo—.

Grignon: Yeah! It was Pixo. It was the original Pixo software. And it was an iPod, that made phone calls. And it was a little wider and fatter. It was like about, you know, it was a little fatter than the iPod at the time. A little longer. It had a Wi-Fi in it. We didn't have any brow[ser]—we didn't know what [the] software [would be] at the time, but you know, we built it! And we had this big fight between Fadell's org and Scott Forstall's org. We had this big fight on what's the thing that we should make? Right? What's the software architecture? Tony wasn't viewed by Steve as a guy who knew how to write software. And that—as much as he did, and he had people that knew how to write software, he was just tinted in Steve's eyes as, "You don't know software. Forstall knows software. Forstall doesn't know hardware. You know hardware." Like, that was how Steve, in his mind at the time, that's how it was set up. And we had this idea to build the phone with—on top of Linux and using WebKit. And we had prototyped it, it was well within like this realm of feasibility. And the argument was always, "We can't ship—we, Apple, can't ship a device based on Linux." And we're like, "Why? Does anybody know what the operating system of the Airport is? Which is VxWorks, but does anybody know?" "Nope." <laughs> "Why should the phone be any different? Do they know what the operating system of the iPod was?" "Nope." <laughs> So, the phone in our eyes was just another version of consumer electronics. And Scott Forstall had promised the ability to deliver a phone based on Mac OS [X], on Darwin, specifically, the kernel. And it would only take six months' additional time. And we would start to be able to leverage the story, which ultimately became the phone. And we had a big knock-down, drag-out fight. And obviously you know who won. While Tony was still in charge of making the hardware, Scott was now in charge of the software strategy. And so, we had this weird—this is where the political game got notched up a few levels, and that fight between Forstall and Fadell was pretty epic. And I was in the middle of it! And that was kind of—because I still counted Scott as a friend, you know, Forstall as a friend, and Tony was a friend. And you know, but company politics, that's what happens. Jeremy was sent in. I used Jeremy to help write the phone layer. The thing that actually made phone calls, the telephony stack. And Jeremy, at the time, was acting as a double agent to Forstall's org. I didn't know that. He actually fessed up to this many years later. Or maybe it was a year. After we shipped the phone. And he fessed up, he's like, "Yeah, I was just feeding all this back to Forstall's org." <laughs> He was like, you know, spy versus spy, but with nerds. He was really bad spies. And so, obviously we ended up with the phone as it was, which was Mac OS [X] based. When you look at the phone, iPhone, it should never have succeeded as a product. And I mean that from a technological perspective. Not from a product perspective. And the reason why that is, is because when you make a new thing, you're going to make a new tablet, or a new laptop, you start with a laptop that works, you change a couple of things, a new display, whatever, it's got this new feature, you're done! Here's your tablet! The phone, we changed every variable we could, and it started with the chip. We invented our own piece of silicon. Not because we could, because we had to. Darwin, as a kernel, required these really high resolution timers that no other chip on the market could do. And so, we had to go to—I think it was

Samsung at the time. We tried—we evaluated a few vendors, but we brewed our own chip from scratch, mostly to accommodate this thing from Darwin. And while we were at it, we wanted to swizzle in [a] much higher graphics core. Which was PowerVR, and very famously, Apple recently said, "We're done with PowerVR," and their stock tanked. <laughs> But we swizzled in all of the things that we wanted, because we can—well, you're making a chip, you may as well make it the thing you actually want. And we had lot of unique needs of moving a lot of memory, a lot of graphics around. So, we had extra wide busses to move huge volumes of data around. So, we started with the chip. Then it moved up to, we had never built a product that did wireless connectivity like this in this particular form factor. The original design, you know, obviously it was metal, well, glass in the front and back, but there was that metal ring that Steve added. Or plastic, I'm sorry plastic on the back, and Steve added this big ring of metal for whatever reason. But that threw all of the antennas off. So, now all of a sudden, we're making a product we've never built before. It's a phone. On a chip that is custom, that has bugs. On an operating system that had never been ported to ARM—which was the instruction set—before, so we're using a whole new tool chain, right? So, this was just as much work as we went from PowerPC to Intel, that same transition from Intel to ARM was just as huge of a deal. So, we had a new operating system, we had a new tool chain. We've got a whole new software stack. The original idea was that we could use some of the original core of the apps that we had—the Mail app, you know, the calendar. We could use the core of those, and we would just rewrite the frontend. That proved out to not work. So, we had to rewrite all of the middleware. We had to—

Hsu: So, the original plan was to actually move the AppKit over to the iPhone?

Grignon: That was the original plan. We might have to do a couple of things to accommodate the look and feel, like for fingers instead of pointers, but yeah, that was the original idea. And that was very quickly tossed out the window. So, now we've got everything changing. The apps are different. They have to be custom. We've thrown in fingers as the input mechanism, instead of a mouse, which was actually a pretty big difference. We had never built a keyboard before. So, autocorrect was not a thing. We make fun of it now, but when you're actually inventing all of this, when you actually see—the way the touch stuff works is when you mash your meat sausages onto the screen, what gets reported back to the software is effectively an oval with a direction. There's a center. But it kind of gives you a rough thing, right? And you swear you're hitting that A key, you ain't nowhere close to the A! Like, the oval that the software gets is somewhere like an inch-and-a-half off. And you're like, "How the hell is this thing supposed to figure out that you intended to press an A?!" Well, it does that with a lot of really interesting early, you know, A.I. was part of that, which was trying to intend what you intended to press. So, as you went A-N, okay, that's maybe a D is next? You know, but again, it's going based on these weird ovals that are being reported. It has no idea, right? So, we had every layer of the stack custom, different, new, full of bugs. And it should come as no surprise that—by the way, with a deadline that was immutable, right? <laughs> This deadline couldn't change. And Steve was betting the company on it. And so, you can imagine the amount of pressure. Imagine you're an engineer in this morass, right? And you go to Mail, you check the thing and the phone reboots, which happened all the time. Whose bug is it? Is it the operating system? Is it Mail? Is it the tool chain? Or hey, maybe it's the bug in the silicon, which happened. So, silicon, when you make a chip today, or any day, silicon's just a bunch of software that gets compiled into a piece of metal <laughs>

and silicon. <laughs> That has bugs in it! And so, it's not a surprise when you have to spin the metal. It's called a metal layer, and you add a bug fix, which is actual metal on top of the rest of the chip. All of that compounds into, "How did this thing actually happen?" there were several times during the development of iPhone, where the program came to a screeching halt, because we couldn't figure out how to move forward. And these weren't like little things. These were program-stopping problems. And these were the kinds of problems where we would have to bring in the people that wrote the software that made the chip <laughs> from different companies into a room with our hardware people to debug these kinds of problems. And Apple, I think is, if not at the time, the only company, I think, that could have pulled it off. Because these vendors thought that they were, you know, being part of the next iPod. Which was a monstrously successful product. So, that's the only reason we got the people that wrote the software that made the chip! <laughs> Because the thought this was such a big deal for the next iPod that they would do that.

Weber: And the kind of problems are what? Just typical crash bugs? But you couldn't tell where they came from?

Grignon: Well, no, actually, one of the worst ones, one of the worst ones was we had, of all things a problem with what's called a UART, and the UART is the serial block of a chip. It's been around since the dawn of chips. It's the oldest thing ever. It's a serial line, and ours had a problem with it so that the connection between the chip that made phone calls and the main chip that ran all the software in very certain circumstances but not as rare as you'd think, that link would go dead and that would result in effectively a dropped call with the chip that made the phone calls rebooting itself, and so, you now couldn't make a phone call until it came back up, and that happened often, often to the point where after we shipped the phone, I changed my license plate on my car and it was ZROBARS, Z-R-O, bars because of the <laughs> number of times during the development that you went to zero bars, which is an indication of a chip crash. But we couldn't figure it out, and you'd have a guaranteed reliable link between these two chips and it's failing. Why? We had rooms full of people, like 30 people who would single step through to see what the hardware is doing. Register by register, it latches this, it does that. <makes sound> Nobody could figure it out. And it turns out, later on—we actually shipped the very first phone with this bug in it, but we invented a—we took a piece of the Bluetooth stack that—I forget the—HCI, something like that. It was a piece of the Bluetooth stack, 'cause Bluetooth radios give you a serial port but it's over a lossy connection, so it's effectively a UART over wireless, which—so Bluetooth solved that problem of a lossy UART, which is supposed to be a hardwired thing. So, we implemented that between our main processor and our <laughs> phone processor so that when the UART shit the bed that layer kicked in—

Weber: <laughs>

Grignon: —and it kept the link alive and it effectively did a reboot but the chip didn't restart itself, and that's how that happened. But it's <laughs> those kinds of bugs, right? You're like, ugh! <laughs> So,

again, it goes back to all of the things that could go wrong did. We had this one problem, when <laughs> we were about to introduce the phone, so this was at Macworld 2007, January 2007. And so, it was around the holidays, around Christmastime, and we had just gotten our final batch of phones from Asia, so they were all hand carried in airplanes in steel cases, and people would go on a—they would fly to China, hop on the—you know, get the suitcases of phones, and not even an overnight stay, hop right back on the plane and fly back. And we had just gotten our latest batch of phones. We'd been setting up for all the demos, we'd been going through rehearsal after rehearsal after rehearsal, lots of problems every day just trying to debug things. Well, we had finally settled on the last set of software for the introduction, so the next rehearsal was gonna be onstage at Moscone Center. Our phones came in and fresh off of the factory floor from China, and we had some bugs in my team that we had to fix, so we fixed them. We did a quality check on our phones that we'd all been using. Everything was looking good. QA, everybody signed off on it. And new phones come in, time to load the new software on these new phones, and somebody had started—we used a gang programmer that did I think 8 or 12 or 16, whatever, a large number of phones at once all kind of plugged in. And these phones now—you had to understand, when the phones come in, you have this big briefcase of things, right, had plastic sheets on them and whatever. Steve Jobs and Jony Ive would put jeweler's loupes on and they would put gloves on and they would grade the quality of each phone, right, based on the kind of defects it had, because when they were still bringing up a line, plastics don't sit right. They're not meshed or they maybe have a scuff during some bad assembly process, so they're given a grade. So, double As are effectively production level devices. As are really good. Bs, Cs, and Ds fall—like Cs basically just go to QA because they're visually problematic. <laughs> So, we had I think eight double As and then I forget how many units we had that were As, and they all went into this batch of gang programmers. And we put this new software that my team had done down onto the phones. And so, they go through the programming sequence and they reboot, and all of them as they're coming back up, they crash, crash, crash a bunch of times, and then finally they all go dead. So, turns out we have a security feature in the phone, the chip that makes the phone call part of things, the baseband, it and the main processor kind of vie for who's the boss. In this particular model, the chip that made the phone calls was the boss, and the security feature in this chip would detect if somebody was tampering with it by squirting their own code onto the chip that would maybe subvert some security policy of the network, maybe make free phone calls—I don't know, whatever your thing is—the security feature existed to see if it had been tampered with. After a certain number of iterations, which happened very quickly, if it detects that I've been tampered with, the security feature literally lights a piece of metal on fire inside of itself and it renders it useless. So, it is a fuse that is blown deep in the heart of this chip, and since that chip is dead and since it's the boss, the phone is dead. It's not like you could just lift that chip off, put a new one on. The whole phone's dead. So, when people say, "Oh, I bricked my phone," right, "'cause you put some new software on it," or whatever. You didn't brick your phone. The way you brick your phone is by lighting that piece of metal on fire—

<laughter>

Grignon: —deep inside the chip, which was instigated by my team, so we burned up our double As, and some of our As, the best phones that we had. These were supposed to be the phones that Steve is onstage holding that were flawless, the best that we could produce at the time, and we lit them all on fire,

well, that little tiny piece of metal. And I thought for sure I was gonna be fired after that, because what turns out happened was because these phones came off of the factory floor from China, we just laid new software on them, there was a bug in our code that was triggered only when there was an existing software on there already. So, it worked fine if you were just a QA person that had a previous version, you flashed a new build, then it was fine. But if you were taking a brand new piece of thing off of the floor and you put the software down, that's what triggered the security thing, which burned the wire, which—so you'll see in some of the pictures at Macworld, there's these glass domes and they're really beautiful phones, and they're spinning around and they're off. Well, those are the best ones we had, and since there was high resolution photography, people could get up close to these things, we wanted those still to have that perfect—you know, the best that we could produce. They were supposed to be running a demo loop <laughs> of just stuff that the phone—like a video that would just play over and over and over again, and a lot of them are off. And you'll see in some of the photos, <laughs> which are basically the black screen and they're supposed to be running a thing because, well, we had lit them on fire.

<laughter>

Weber: So, how much had they gone with more off-the-shelf operating system, less custom hardware? Presumably this would've been a much shorter, simpler—

Grignon: It would've been an easier effort; however—

Weber: Would the results have been—

Grignon: No, I don't think so.

Weber: —very limited by comparison.

Grignon: It was, and we go back to that. You know, we'd have to move lots of data around because Steve was adamant about the fact that this thing couldn't dip below 60 frames per second on the animations. Most of the animations scrolling the list was one of the big things, clocked in at 90 to 120 frames a second. But if you went below 60, the way that it works, the video chip and the refresh works, is that would then degrade to 30. So, it's not like you get 57, you get 30, right, so at the time it was a very visible difference, and so, Steve wanted well above 60. So, those are not a constraint that most operating systems at the time could handle. There was a lot of work that Apple did, that these teams did cross-functionally to make all that happen, and that was pure invention. That wasn't like, "Eh, we're gonna tweak this and do that." No, this was invent new ways to do things. It was fascinating.

Hsu: I was curious, so you had worked on Dashboard. How much did Dashboard influence the design of the iPhone?

Grignon: Not much. So, the UI team was given creative license without—whatever you want to create. Now, the funny thing is Steve wanted a phone based on the phone we should build, not the phone that we could build, given the constraints. And so, a great example of that is—so we weren't allowed to hire people that had made phones before, so when you're making a phone you're like, oh, well, how am I supposed to build a phone if we don't know how to—well, you can't google it. You have to invent it. Well, we're engineers so that's what we do. So, you deconstruct the problem just like anybody else does and you start to kind of build it up from scratch, with the idea of what you're actually trying to achieve, not what we can achieve. So, I was the guy that had to pitch Visual Voicemail to AT&T. And Visual Voicemail obviously is a feature we kind of take for granted, but back in the day, you had to hit like seven to go forward, four to delete, whatever. You had to hit—you'd remember the numbers in your head to navigate your voicemail, and Visual Voicemail was kind of a key feature that we had. And when you're a software engineer working on Mac OS [X], Visual Voicemail is very obviously [like] the Mail [app]. You know, the attachment is the message and the metadata around it is caller, all that kind of stuff. And we can represent it that way visually. So, I go to pitch Visual Voicemail. It's this awful—it was basically like a terrible wedding. <laughs> Like, there's this room, a reasonably sized conference room and there's Apple people over here and AT&T/Cingular people over here. And the woman that was running—she was the CTO at the time, was right in the front, super nice person. And we go up and I give this, "This is Visual Voicemail. This is—you know, we can't show you the actual pixels, but this is the idea. We obviously need help on the back end because, you know, we don't know how that works, but here's how great it is," right? And after my little impassioned plea, she all but pats me on the head and says, "Thank you, but this is how complicated it is and here's the rebuttal. This is why we can't do Visual Voicemail." So, this guy gets up and he starts talking about how immensely complicated voicemail systems are. You know, "They've been doing this for many decades and there's this, the sphincter valve and the McNewton nut, and there's like six people in Iowa that have this very specific concern," and whatever, he just proceeds to say all the reasons why you can't do this elegant design. And we argued about that for a while, but at the end of the day we were at a stalemate. And so, I went crying to Steve. I was like, "Look, we can't do this. <laughs> They said there's the McNewton nut and it's complicated and six people in Iowa..." And Steve went to the CEO of Cingular and made it all better. Turns out, that single act was a concession that gave them, AT&T—it was AT&T and Cingular, 'cause that's what they acquired—that single concession gave them the exclusivity on iPhone for five years so that they would do Visual Voicemail. That's how important this feature was to Steve. And turns out, like two months later, it's working! Huh. Lucent already has a back end that's been tested, and it was just a game of showmanship. But could you imagine if I had hired on my team, our brainstorming team, somebody from Nokia or from Simbian or whoever, you know, people at Panasonic, people who had been making phones, if I'd hired that person on my team and in our brainstorming session we were like, "Oh, hey, it's like Mail, right?" If this person would've been like, "Oh, you guys don't understand, right? There's the McNewton nut. There's the sphincter valve, and these people in Iowa..." We would've torpedoed the idea before it would've left kind of that brainstorming session, so that's what Steve was hoping for when he dictated "thou shalt not hire people in the phone business." It was because of the McNewton nut and the sphincter valve, right? We don't want to build a

phone that we can. We want to build the phone that we should. And some of the times you've got to run right smack into the wall, and then he was always there to kind of make it better and fix it. It was a unique thing that Steve had. But it taught me a really interesting lesson, right? There is—I wouldn't say groupthink—but we naturally kind of impose restrictions. When you're trying to innovate on a new thing, you taint your thinking just naturally, and it's very difficult to step out of that box, so to invent a new product you have to distance yourself from existing thinking and you have to invent, and sometimes that means running right smack into the wall, and you have to. <laughs> Otherwise you're gonna end up with the same thing, just iterations. And I think that's why—you know, when we started iPhone in earnest, we were all carrying the Palm Treo 650. It was the phone that was, whatever, Palm OS based. It had a little stubby antennae on it. It had a keyboard, a resistive touchscreen, not a capacitive one, and we were carrying that—that was the baller business phone at the time, right? That's if you were anybody you were carrying that phone. And we used that phone as a reminder constantly of the product we didn't want to build, which was funny because I ended up going to Palm—

<laughter>

Grignon: —after that. So, we ship the phone, right? Everything goes out. It's great. Rubinstein calls. He's now left Apple. He's off doing whatever, we didn't know, and he calls up and he's at Palm. And I didn't know this, but he's about to be the CEO of Palm and he's like, "Hey," <laughs> basically, "Do you want to come to Palm?" And I'm like, "Hey, no. <laughs> Don't you remember, we all carried the Treo as the thing we didn't want to end up with?" And his argument was, "Well, look, you know the roadmap at Apple, right?" We planned our roadmaps out 18, 24 months in advance. They're all in this big war room. They're all carefully plotted out on these long timelines. He goes, "You look at that roadmap, what is it? It's iteration. There's nothing new," which was borne out. I mean we saw the product roadmap come out, you know, the iPhone 2 and 3 and 3S. He was like, "There's nothing new there." He goes, "I want to build the phone that we set out to build initially." He goes, "I've got a brand new design team. We've got some crazy concepts on the graphics and the art, the interaction, but we need a software story," and that's what Palm turned out to be. And so, I was like, "Yeah, you're right. I could sit there and turn the crank and make a lot of money at Apple or I could do something new and invent a new operating system." And that ultimately became WebOS, And so, WebOS was Dashboard. When you ask about the influence that Dashboard had on iPhone, it didn't. Dashboard had all of the influence on WebOS. So, Matias Duarte who was the designer at the time of Sidekick, Danger, the little phone that had like a little rectangular slab that the screen kind of swiveled out and had that color wheel that could be programmed to do all sorts of stuff. Matias was—still is—a very talented designer, and he was kind of disenfranchised with where Danger was going, and Rubinstein had approached him about creating—"What's the operating system, or what's the look and feel of a phone that you would create?" And he had this really gorgeous design, but there was no way that the existing story, the software strategy that Palm had in place could execute on that visual design. He had all sorts of crazy stuff, cards and lots of things running at the same time, and they're all supposed to be live and all sorts of crazy stuff. And when I got to Palm, I saw that the software strategy was untenable. Like, that was not—we were not on the path to success. And so,—

Hsu: Was Palm using BeOS?

Grignon: No, actually Palm was using Palm OS. It was its own thing.

Weber: Okay, but when was that sold to Access?

Grignon: Oh, god. Palm went through PalmSource. I mean they had several iterations of the company, And so, they—

Weber: But when you came they had already sold the original Palm OS to Access.

Grignon: To Access, and then who owned that, SoftBank?

Weber: Access?

Grignon: Uh-huh.

Weber: No, they were independent, I thought.

Grignon: SoftBank was—like they invested—like the money behind it.

Weber: Right, but Access—

Grignon: And they were their own thing.

Weber: —they also did i-Mode earlier on.

Grignon: Yes, they did. At this point they had folded everything back in, and the people who were part of Access were then part of Palm—'cause a lot of them I had on my team. So, I get to Palm and we couldn't center a block of text without having somebody write the code to center this block of text. I mean it was just such an infantile state of affairs, things that you take for granted when you're building—like as basic building blocks—we couldn't do. And going back to what we talked [about] earlier, Greg Simon, the guy that wrote NewtScape for Newton, I walked down the hallway one day and he's there, and he was a good friend of mine. I was like, "What the hell are you doing here?" And he was like, "What the hell are you doing here?" <laughs> It was kind of one of those things. And we just started talking about—you know,

he'd been working on another thing, all part of Palm, and he and I, I remember we were grousing one day about just the sad state of affairs with this operating system that we were on. We loved Matias' vision. We loved the hardware design that was en route, but we weren't gonna get there. And so, we started kind of talking about Dashboard, and I said, "You know, on iPhone we had this idea of using WebKit and Linux," and we already were using Linux, a flavor, a variant of Linux, on this Palm thing. Like, they were [a] competing effort. It was McNutt or something like that, whatever. It was some weird thing. And what if we just took all of the underpinnings we used for the display layer—WebKit, 'Cause you don't have to write special code to center a block of text. You get alpha blending for free. <laughs> And over the weekend—things had started to come to a head organizationally. Nothing was working and gelling. And we crafted, he and I put a demo together. It was like a Friday afternoon we did it, and then Monday morning we'd pitch it to the CTO at Palm. And we took the artwork that we were given from Matias and we had done more in a weekend using WebKit, effectively using Dashboard, what Dashboard was founded on, native code underneath, using WebKit as the display engine, we did more in a weekend than we had as a 450-person organization created over months and months and months, and it was a very seismic shift. It was like, whoa. We can actually do this. Obviously I had a friend in Rubinstein. He'd brought me in. He wanted this vision all along. And we had a pretty tense argument about where to go with all this, and ultimately I was given the green light to pursue it. And so, I picked nine people, myself was ten, so I put ten of us into this conference room, and kind of drawing from those old early Apple days, I had ten people for 30 days, and let's sketch out the ideal operating system. I'm cancelling all of your meetings all up your management chain. None of this—what matters is the next 30 days. I had some black sticky paper put over the windows, just the typical pirate-y kind of thing that Apple was notorious for back in the day, and it was interesting. So, in 30 days we created WebOS, the basics, the beginning part of WebOS. In that 30 days, I inadvertently also created a culture of have and have nots. So, you have—I mean everybody knows—the same thing happened on iPhone. iPhone was notoriously secretive. We put up walls and doors for the very first time in Apple's history. There were all of a sudden parts of campus you couldn't get to—aside from the industrial design lab at Apple, which was expected because that's all the secrets—all of a sudden where iPhone was happening—but people didn't know we were building iPhone—were these big glass doors you couldn't get through unless you were badged in, and that was the initial have and have nots, and everyone sees who's going in there. You know all of a sudden this person you've been a peer with is now all of a sudden behind the mysterious glass doors, and what does it do? It makes people feel bad, right, but more importantly it creates focus. And so, I did this for Palm. I put ten people in this room, and if you looked around the company, it's like, well, these are ten really sharp people, and Matias and me and all these—we're just focusing in, and that created a lot of animosity, but we managed it I think pretty well. So, in that 30 days we now have a thing that was WebOS. It was a path to market that would get us a product in one year. So, from the invention that weekend, to when we announced it at CES and we got Best of Show, was one year. And <laughs> so the CTO, the same guy that blessed all of this to happen, I asked him kind of around this period of time after we finished, I'm like, "Hey, I want to take the team out. I want to do something special for this group of men." He was like, "Yeah, go for it. Like, yeah, this is great." In that time, I had a kid, my second child, my son, and in that time I also had a new boss, this guy named Mike Abbott. Mike was hired to be the new SVP of software. He ran all of software at Palm, the cloud piece, my piece, all of it. And so, I had a new boss in between all of this. So, Abbott's my new manager, but I'm out on paternity leave. And I come back, take the team out. We went to this steakhouse down at Alexander's in Cupertino, and 10 of us had the best time. We still talk about it

to this day. It was anything on the menu, all the drinks. But, again, we're nerds. We're not like wine snobs, but we ran up, for 10 people, [a] $5500 bill, cigars, all of it. But, you know, CTO said it was okay. And so, I go in, hadn't really met Abbott at this point, and I give my receipt for the dinner to his admin just as an expense, didn't think anything of it. Like, two hours later she was like, "Hey, Mike Abbott wants to talk to you." I'm like, "All right, great," so I head on over. And Mike sits me down, our very first real meeting. Like, <laughs> we hadn't even done a one-on-one. And he was like, "Look, I've got the fiduciary responsibility to have this conversation with you. It's not okay to have $5500 dinners." <laughs> I was like, "Well, but I respectfully disagree. What we did in 30 days was we changed the course of this company, and we effectively gave this core group of 10 people a $550 bonus for altering the course of Palm. What would've happened if I gave you a check for $550? Well, after taxes it'll be, what, 300 and change. What would you do? You'd kick me in the nuts and quit on the spot. <laughs> Like, thanks for nothing, right? Instead you got this experience that to this day when we get together we talk about that dinner." And I think that was an interesting management lesson for me, right? So, it was, like, "If I just give you another couple of bills that show up as just more numbers on your paycheck, that's easily forgotten. But if you create an experience and you start to do something really special, that's where that impact gets made." And so, that for me was a really, really huge lesson. But back to Palm. You know, we all of a sudden have this problem now of the 10 people that I had stuffed in this room created this cool thing, now all these people that were pissed who were working on this other thing in limbo and whatever, all of a sudden we need all of their help to make this happen. So, it was an interesting management challenge but we did it. People rallied behind the vision. We worked nights and weekends and all of the things. All of that led up to I think probably one of the coolest product announcements at CES, and we were gonna be showing off our new phone. Coming up to it, I remember showing my friends at Apple before—we'd done all the rehearsals and all the stuff and I went out with my iPhone buddies, Ganatra and Hertz, just having a dinner, and I show them the prototype that was about to be announced like in two weeks. I was like, eh, it's a little bit of a risk but screw it.

<laughter>

Grignon: And I could see they were like, "All right, this is..." I mean they were friends so they broke my balls and talked about how crappy it was. But at the end of the day I got them to admit, they were like, "That was a legit—we were concerned, still are." And actually that was the only time—so Steve and I had a falling out. I'm not allowed back at Apple, but that was one of the few times that Steve actually—the only competitor that he really gave a hat tip to, and he actually in some article said something like, "This is an actual legitimate competitor." And so, it was kind of a nice feeling, to have that. When I left Apple, I'm kind of going back in time a little bit, but when I left Apple I gave my notice, after I'd told Rubinstein that I'd show up and do this thing, and I gave it to Fadell and like two or three hours, like an hour, whatever it was, Steve's admin called and she was like, "Steve wants to talk to you this afternoon." Great. You kind of expect something like that after everything that we've been through. And I show up to Steve's office and he had a little conference room right next to his actual—that he did all of his kind of one-on-ones in. And so, I sit down, and he was always on time except for this one. He wasn't, right? I'm sitting there in the room. It's like five minutes, ten minutes goes by. I'm like, okay, so I'm kind of figuring I should just go? I don't know. He comes in, sits down in his usual spot, and kind of sits down and he looks at me for a little

bit. There's no pleasantries. There's no, "Hey." There's no whatever. First thing out of his mouth is, "You know you fucked up Bluetooth on the phone." I'm like, this is the keep me talk? Uh… <laughs> And so, I was like—there was this, what do I do? What do I say? And I'd already made the decision to go to Palm but he didn't know that, and so, I was like, "Well, I disagree," and I defended myself. I was like, "We had vendors that were completely in the dark about each other," and I kind of went through the technical reasons and he kind of, "All right. How long have we been working together?" and he was like, "Back in iChat," and he starts kind of recounting this little history, right? He was like, "Five years. You've been a member of my family for five years." And he starts kind of going off on this lovey-dovey, right, like now the keep me part starts. And, "You know, I hear you want to leave. There's lots of other projects happening. Why do you want to go? We can do—let's arrange this, that, and the other." And when it was clear that I wasn't gonna bite, that I had made the decision to go, then it started to take another turn dark and he brought up his family. He was like, "You know, if you take so much as one member of my Apple family away from me, I will personally take you down." I'm like, "All right. That's fair." <laughs> What do you say to that, right? And we just had this up and down rollercoaster. And he was like, "You know—" and then it goes up and then it goes down, it goes up. And finally, this is like 40 minutes in and he says, "I think we've got a great future ahead of us. Let's meet tomorrow and we'll go through some options." I'm like, "Oh, yeah, great." <laughs> I had no intention, and I didn't meet with him again. I just left. I didn't have the balls to actually say, "No, I'm done." I just said, "Yeah, okay," and then I just didn't show up the next day. Fast forward now, so I left. I actually got escorted out the next day after I failed to show up, so I was terminated on the spot, even though I'd given my two weeks' notice. It was a very Apple maneuver to do. Fast forward to WebOS. I've now got the basics up and running. I've now introduced it to the executive team. This guy named Roger McNamee was Palm's biggest investor. It was Bono's investment arm called Elevation. And I'm pitching WebOS—not pitching. It's already a done deal. This is the direction. So, it's in an exec offsite. So, I'm showing off WebOS to McNamee and everybody, like, the exec leadership team, and they're like, "Yeah, this is great." Afterwards, Abbott was like, "Hey, I want you to go meet Roger. I think you guys would hit it off." And so, he's busy talking to a group of people and we kind of get in there and he's like, "Hey," we do the intros, "I'm Andy." "Great." He continues on his story, and like 10 seconds later Roger stops himself. He goes, "Wait, you're Andy from where? Andy from Apple?" I'm like, "Yeah. I worked with Jon [Rubinstein] and we did iPhone and this, that." And Roger, very warm, kind of—he comes in and he says, "I want to shake your hand," and he kind of brings me in really close. And I was like, obviously I got recognized. He knows who I am, right? And he goes—weirdest conversation I think to this date that I've ever had—he goes, "Let me tell you what happened the day you left Steve's office." And I was like, "What?" <laughs> So, Roger notoriously had like four phones on his belt. He had a Palm phone, an iPhone, a Blackberry and a, whatever, an Android. He had all the phones. He goes, "I've got all these different phones, all these different phone numbers. My wife Ann is the only one that knows," and he had, whatever, his iPhone. And he was like, "My wife Ann is the only person that knows this number. My admins know this." He goes, "When you left, all four phones lit up, and it was Steve and his assistants." It was all of his assistants. It wasn't Steve on the phone—"going after me," Roger, right? And there was a weird thing happening between Rubinstein. Steve wasn't allowed to talk to Rubinstein directly because of Fred Anderson and Elevation, and all sorts of weird things happening with Apple's price—not price fixing—

Hsu: The stock thing?

Grignon: The stock thing.

Hsu: The stock backdating controversy?

Grignon: Yeah, it was backdating. And Fred Anderson was a partner over at Elevation. Rubinstein was involved somehow, and so, it was all this weird—so Roger was the only person that Steve could contact at the time. And he said all of them lit up and he goes, "I didn't know. I wasn't gonna touch anything. So, I found out later it was because you were coming to Palm." And it wasn't me, so Steve wasn't pissed or afraid that I was coming to Palm. I'm one person. I'm replaceable. Everybody knows that. He was afraid of what I did and was about to do, which was take 20 people from Apple into Palm. And so, now we get into this collusion effort between Apple, Google. We're not gonna poach each other's people, Facebook. I mean there's all these things. Palm actually very famously—Ed Culligan, CEO at the time, and it's in the lawsuit record, didn't agree. Steve was like, "Let's agree to not take each other's talent," and Ed's like, "No. <laughs> I'm not buying it." And that was the beginning and that was exactly—again, it wasn't me. It was what I was going to do, which was me and 20 people, 'cause those are really expensive people and brain trusts to replace. But that's—you know, Palm was right in the middle of all that. That was a fascinating time to be part of the kickoff of the big talent war.

<laughter>

Weber: And when you left you didn't try to negotiate something more at Apple, obviously.

Grignon: No, and I didn't want any more. Apple iPhone—you know, going back to that story about how every variable was changed—from an engineer's perspective, just from—I changed as a person. The management stress was just off the charts, probably the worst professional experience I think I've ever had, just given how difficult it was. And I mean I remember leading up to the announcement in January. <laughs> So, the people that made the chip, the baseband, were based out of Europe, and we had all these awful bugs in their stack, in our stack and nothing was working right, and it got to the point where we're like, "Look, we just need everybody onsite," and I had them bring like 20 or so people from Denmark and Germany and wherever. We flew them out long term to Cupertino and I shoved all of them in this windowless, cold server room. And <laughs> we got to name our rooms and I called it European Vacation, so there's this little placard outside that says "European Vacation." It's just this dungy, crappy, awful room for all these people, and I had a morning meeting every day just to go over our bug list and bug, bug, bug. And I'd started—a lot of us had adopted Steve's temperamental management style, just the explosive, unpredictable, emotional. And I go in one day, so we're going through the bug list. I'm going in, I'm driving in from my work meeting, and I'd just gotten my ass handed to me by Fadell who had gotten it by Steve, and it was just like—I go into this meeting, my morning meeting just super agro and

angry, sit down at the head of this table. There was like 15 people at this table, and I start going through the bug list, specifically the thing that I had just gotten yelled at about. And the project manager sat at the complete opposite end, and we had gotten kind of comfortable with each other, so the early days of him being very studious and just very on top of it, kind of relaxed into a very sloppy kind of way of interacting. And, you know, he was just kind of leaned back and he had his legs crossed and he was like, "Eh." You know, I asked him something about whatever I'd gotten yelled at, and he just took that particular interaction way too lackadaisical, given the just sour mood I was in. And for whatever reason I just lost it and I just—like a crazy person. I'm slamming my hands on the table, banging, and I just feel myself losing it, right? You know, that's not me, but that was the cauldron, that pressure cooker that we'd been put into. And as I kept going on and on I just kept running away from my grip on reality. And it culminated with—I had my laptop open and I was just screaming and I'm beat red, and I take my laptop and I threw it against the cinderblock wall that—this server-less room, and I just, <screams> whatever. In my head it was like an explosion. Like, it hit the wall and <makes sound> parts are flying everywhere, and it was, like, this big, climactic scene, and then I storm out of there. And I'm sure really what happened was it just kind of like, thunk, <laughs> slid down to the ground, very anticlimactic, and then I just stormed out of there like a weirdo. <laughs> But that was one of those moments where you just realized just that insane pressure that everybody was under, the reaction to the pressure. The way that you handled it, all were good lessons because, again, that's not me, but I saw that in a lot of people. We all kind of lost that grip. And I think when you look at a project like iPhone, you know, it takes a pretty significant personal toll, professional toll. I mean obviously it was a successful product, but there was a lot of sacrifices that a lot of people made. There was a very famous story, was like December 18th, 20th, whatever, sometime before Christmas, and late night in Scott Forstall's hallway. I was hanging out with people over there, and it's probably like eight or nine o'clock at night, and we're in the hallway, just whatever. A few of us are hanging around, we're debating, and this person comes up, and the guy I'm standing next to and her kind of get into it about just whatever, some bug, it's just whatever, nothing big, starts escalating very quickly like in a heated exchange. And I forget who said what, but it was basically like, "Well, if you don't want me to see my kids this weekend at their holiday show then I guess I can handle that now." It was some kind of quip like that, right? And this other person had kids and they're like—it became this surreal debate about who spent less time with their kids. <laughs> And it turned into this screaming match between the two of them, and this other person storms off so angry, goes into her office and slams the door, just hard, like boom, like this huge, you heard it all the way down the hallway. And we're like, wow, that's pretty—so we all kind of walk down there to make sure things are okay. She was trying to get out of her office. She slammed the door so hard that the lock mechanism broke <laughs> and she was unable to open the door handle. And then it turned from this explosive thing into like, oh, shit, we've gotta get her out of this room. Oh, my god. And we're, like, jiggling it and there's a glass panel. Apple has solid wood doors and there was, like, a glass thing and then there was your office. And <laughs> it turned funny very quickly. We're like, how do we get her out of there? <laughs> And we call Apple locksmith and they're like an hour away and it's late at night. And I forget how he got there, but Forstall had been working late and Forstall shows up. He had an aluminum bat and we're all taking full-size whacks at the door handle, at the lock on this door, just like boom, trying to get this thing unstuck. And finally it was Forstall who just went agro on this thing and he beat it to the point where it actually popped off. The whole lock just popped off and then we got her out. But that is, like, it was a moment where it started off as first a really surreal and weird debate about who spends less time with their kids, into this explosive moment, to a very funny but concerned,

like we've got to get her out of there, to this cathartic, all taking a whack. It's like that scene in "Office Space" when they're all just beating the crap out of the copy machine or the fax machine or whatever it was. <laughs> It was that moment—and none of that was lost on any of us, but that I think is a good indication of just <laughs> how tense it was.

Weber: And the glass wasn't big enough for her to come out?

Grignon: No, it was, but we didn't want to break the glass—

<laughter>

Grignon: —and it was more fun, by the way, just to hit the lock with a bat.

<laughter>

Hsu: I've heard a story where you actually went to Steve Jobs' house to debug the radio on his iPhone.

Grignon: Yeah, that was very weird. That was also leading up to the announcement, the introduction. And we had this weird problem where Steve had this black hole of a house. He has these really thick walls, like four feet thick brick walls in his house, impenetrable by wireless anything, right? And the AirPort team knew this, so the AirPort was the first team that had to deal with Steve and his crazy house. Well, it was our turn, and he had this weird scenario we couldn't reproduce. Whatever he was seeing we couldn't reproduce in any QA setup, any lab, any whatever. And finally I was like, "I need to bring some people and debug it at his place." And I was setting it up, so I get in touch with his admin and I'm like, "Look, we need to do this." And everything is finally—so he's calling me to arrange a time. And so, Steve had called me a bunch of times, but whenever he called, caller ID at the time said "Apple Inc." or from his admin or whatever, and this one day, it was Saturday morning, and my wife—the phone rings and she picks it up. The caller ID—she's like, "Oh, this is for you." It said "Steve Jobs" on the—<laughs> he had one line at his house that actually had his real caller ID on it, and it was him calling from that line. So, we set up a time like two hours from then, and we show up with—so I get my wireless team, a couple people from the kernel team. All six of us show up at Steve's house and they get to work, right? So, actually we pull up to his house and he had two plots of property and he had dozed the other house so had turned it into a garden or something like that, and then he had his main house. And he had this stubby little driveway with a gate, and his car—he had a Mercedes that he had—very famously he'd leased or just had—no license plate on the Mercedes, whatever, and the car was parked up against the closed gate in the stubby part of the driveway. And we pull up right next to him on the street, get out of the car, and you just hear the radio blaring in this closed car. He had all the windows shut and the door, and he's rocking out in his car just by himself.

Hsu: <laughs>

Grignon: And it's loud enough where you can hear it feet away. We were like, okay, that's different. And so, we get out of the car and he turns it off. He's like, "Hey, guys." <laughs> He just hops out like it's Saturday. Let's debug some stuff, right? <laughs> And very nice, normal experience at that point. And then we go in and people are all doing their thing, and he and I are just chatting. We're going through the phone looking at the day's build, and he's asking me some stuff and pointing out whatever. And since it's Christmas, they have temporary drivers for FedEx and UPS and whatever. And we're just sitting—the back of his house is this huge, multi-panel glass walls, right. All let open up to the outside of the back. And there's a little path that goes out from one of the back doors to the little sideways thing. We're just hanging out sitting in front of these windows going through things, and out of nowhere you get this crazy, archaic <makes sound> buzzer sound. It wasn't like a doorbell. And it scared the shit out of him, even. We were like, "What the fuck was that?" <laughs> We were looking around. And he doesn't know what the hell is going on, either, so I felt good about that. And out of the side glass coming up the door is the happiest FedEx guy you've ever seen, like just kind of striding on in, temporary guy. So, his normal FedEx person just knows to just leave the package at the whatever. This guy hit the weird buzzer, freaked everybody out, and it was the first time I think anybody had ever come within two feet of an iPhone prototype 'cause they were all on lockdown and the guy didn't know it. The guy comes up, has Steve sign, 'cause he knows he's delivering a package to Steve Jobs, right? This guy is excited. Steve had his prototype phone. He just kind of casually puts it behind his back, signs the thing for the guy, gets his package. And the guy was—it's funny because you look at kind of the introduction of this thing and I think that guy is probably the closest any normal came to it and he had no idea. And it's kind of funny in that same vein, when he introduced the phone at Moscone and he had his kind of—the money shot at the end was when he was browsing the web, listening to music, decides to order 4,000 lattes for everybody in the audience, calls up Starbucks and this woman answers and she's like, "You want what?" And he's like, "I'm just kidding," right? They did a follow-up piece with her. They were like, "Did you have any idea that Steve was doing that?" She was like, "No, of course not." <laughs> "I just thought it was some random crank caller." And I think this guy would've been the next one. Like, you delivered a package to Steve and he had this thing right behind his back and you had no idea.

<laughter>

Grignon: And the guy I'm sure to this day still has no idea.

<laughter>

Grignon: One of the more surreal debugging experiences <laughs> I think we'd ever had.

<laughter>

Hsu: Do you have any stories about the keynote, the unveiling?

Grignon: Oh, god, yeah. <laughs> So, we go up to Moscone the day before, get in a hotel room 'cause it starts really early. We did an early morning rehearsal and then we started the event, so we had to be up really early up in the city. And sometime that night we decided that it would be a really great idea that after each chunk of the keynote, whoever was responsible for that particular piece would do a shot. And we had brought up a bottle of scotch, Johnnie Walker, I think, or something like that. And so, we had this 750 ml bottle of scotch and we had made this pact between six of us or whoever. And we were up close in the front watching the demo from the audience, and I realized kind of before then, I was like, oh, wait a minute. I'm actually responsible for all of it because it was stressing the radios, it was stressing all of the things. And somebody else had brought up—he was like, "Hey, you've got to do a shot for each one." Like, "Hmm, all right. <laughs> Twist my arm." And so, there were six of us. We had segmented the demo into six different pieces or whatever, and so, after each chunk worked, like, I'm doing a shot of this Johnnie Walker or whatever. And I'm a big guy but, man, six shots, that'll get anybody. <laughs> At the end of this demo, all six of us finished off the bottle but I was hammered. And then we went out 'cause we had just finished. It was the best run-through of the demo I think I'd ever seen. We had never had as good of a run-through in practice, in rehearsal than we ever had at the unveiling, and so, that was a really interesting stroke of luck. And I remember looking back, I mean we were up in the front and there was this huge wall of reporters and people—Apple's events were spectacles at that point. There was just hundreds and hundreds of cameras, and I remember thinking, these people have no idea what's about to happen, what they're about to see. And to see it, kind of after the fact—'cause in the moment you're just scared shitless. <laughs> It's like, is it gonna work or not? So, I wasn't really paying attention after anything outside of that immediate moment, but then after you have that chance to decompress and kind of let loose, it was like, oh, my god. That was cool. Now, we didn't know, funny enough, I don't think any of us really thought the phone was remarkable in any real way other than it was just a really great announcement of a really cool product. I don't think any of us thought that this was going to change the way people interact, didn't think that it was gonna launch new businesses. It was just gonna be a new thing. But that didn't diminish from that sense of pride you get when you launch any new product, right? It's like that same sense—I remember feeling the same way when we launched iChat or Dashboard or any of these other things. You're just like, fuck ya, that's cool. You had that same kind of sense of pumped-upedness [sic]. And kind of looking back, I've seen—you know, when I go out nowadays you see entire families sitting around a table at a restaurant or whatever, and every one of them is just staring at their phones. And did you set out to create that? No, I don't think any of us did. And if you would've said, "Hey, here's a glimpse of the future. Look at this. This is what you're about to announce. How great is that?" You'd be like, eh. <laughs> I don't know if I want that, <laughs> but, hey, cool, I can get real-time mapping for my phone? That's neat. <laughs> But, yeah, and I think what's funny was when we announced the phone, Steve had a very adamant view on not allowing developers to be part of this thing, right? The phone's job, number one, was to make and receive phone calls. That's it. And the second we let some knucklehead developer in to write a piece of software that can potentially crash the thing, right, because, remember, at the time it didn't take a whole lot to crash the phone. If we were to allow just some developer to write some piece of code that took down the whole device, what happens if you got in a wreck and you need to call 911? <laughs> Like, all of a sudden the phone's not doing its job. I don't think

any of us really ever saw a future where this thing's number one job was basically not to make phone calls. <laughs> I mean when was the last time you actually made phone calls? I mean I do every now and then, but most of the time I'm sitting there on Twitter <laughs> and whatever, like anybody cares what I have to say. You're in line at Safeway, what are you doing? You're on your phone. You're not making phone calls. <laughs> You're not whatever. But that's a very stark difference 10 years ago and today, right? We use our phones for pretty much anything but actively communicating with another human. <laughs> The irony.

Hsu: So, the App Store was not—we know that the App Store was clearly not part of the original plan, but I've heard a lot of conspiracy theories from developers who say, "Oh, they did know but they just didn't want to announce it."

Grignon: Nope, nope, nope. We actually thought we could get away with a Dashboard-esque style developer story where you have access to WebKit and you can write apps in a Sandbox, 'cause we can protect the Sandbox, right? The second we allow you to write native code, all bets are off, but things have changed.

Hsu: But there still is a Sandbox even with native apps now.

Grignon: Kind of, yeah. I mean we still have instances where apps can, in very specific instances, bring down the phone, but I mean—actually, that's a fault of the kernel more than it is of anything else. The kernel shouldn't allow the rest of the device to get into a state where it's inoperable.

Weber: And how much was—I mean web browsing, though, was the obvious major function besides phone calls, right?

Grignon: Well, so Steve—

Weber: Well, music and web browsing.

Grignon: Music, but because, again, we were still riding high on iPod success, right? And so, people still had this really huge attachment to their music, so Steve wanted to capture that. And all of a sudden he introduced it as a thing that made phone calls and a revolutionary Internet communicator, <laughs> which is basically web browsing. And so, he was pitching the idea of a full desktop browser experience inside of this tiny little screen, and that was new, I think, and that's when I actually figured we had a real product on our hands.

Weber: And I mean i-Mode was a little bit of a premonition of that in Japan, kind of the mobile web, but were there any models for that?

Grignon: No, I don't think so. And I think the interesting thing was there was nothing at the time that you couldn't do on your phone that you could do on a desktop, right, and the desktop was still the world class kind of internet experience, and these other things, these kind of crippled web versions that existed for mobile devices were kind of viewed as second-class experiences. And this was, you could have—you want to watch videos, watch your videos, all of it. You just pinch and zoom, and that was a completely different—so that was the—you know, so I've got two records on the iPhone. <laughs> I should actually see if I can get Guinness to write them up because the best part is that you can't break them. <laughs> I'm the first person to receive a phone call on an actual production phone from an actual production phone and the first one to actually look at porn.

<laughter>

Grignon: We had just gotten our batch of phones from Asia for our use, and we had two versions of the software. One was the real UI and one was the UI that my team had made. It's called SkankPhone, and SkankPhone was just this crappy looking device. It was mostly for QA, and everybody who didn't have UI access to be able to exercise all of the features of the phone, you could open up an embedded browser or you could make phone calls, you could more importantly do all the testing that Cingular had to do, but most people didn't have full UI access. So, we got these units and I'd walk a bunch of them over to Forstall's org and we'd load the software up on our prototypes the very first time it's all running. And it was like an hour, we were just all sitting there quietly. I remember exactly, it was this purple couch at the end of the hallway, and we're all just kind of sitting in our chairs and I was on the couch and we're just sitting there quietly for like a half an hour just trying out all the things that it can do. We tried phone calls, we tried—and I had the browser open. And in that hallway was where WebKit was developed, 'cause that team sat right next to me when we were building iChat. And one of the interesting, weird things that you have to deal with at some point when you build a web browser is the natural thing people are gonna do, which is look at porn on it. And it's a weird problem for an HR department to have because all of a sudden you get bug reports. It's like, "Hey, Hustler is doing this and it crashes 100 percent of the time when you do X, Y, Z. Somebody has to debug it." <laughs> So if you have that kind of traditional filtering software on your corporate network, you don't allow any of the adult content through. And because of Safari, because of WebKit, we had all of that removed so we could look at all of the porn. And we had to sign agreements saying we wouldn't be offended if we saw it. We wouldn't pursue Apple, and it was a whole big thing. Anyway, that's where that happened. So, we were sitting there in the hallway and we were just going through stuff, and I was like, "You know, I wonder what porn looks like on the—" so I went to this website called Foobies. It doesn't exist anymore. It's just boobies, anyway, whatever. And so, I went to it and I'm like, "Oh, my god, look at this guy." <laughs> So, we dorked around with that for a little while, and I think like two weeks later IT shut it down, access to that site or whatever. But that was the first porn site ever viewed on an iPhone.

<laughter>

Grignon: I don't know if that's one that you should really brag about, but, you know, somebody's gotta.

<laughter>

Weber: <overlapping conversation>

Grignon: Oh, and the phone call was funny because we had gotten these units back and they had loaded up the software. My team obviously was building the telephony layer, and I was in a meeting at the time up in my office, and this was in Mariani One and they had just gotten the software stack up and running with SkankPhone and all that. And I'm talking to somebody and I get this phone call. And I look over on my phone and I'm like, oh, I don't know who that is, so I sent it to voicemail and I finished my meeting. And I checked my voicemail afterwards, and this was while I'm meeting with this person. My guys are like <makes sound> rapping on the thing. I'm like, "Go away. I'm actually doing a real meeting here." So, I check the voicemail and they're like, "Dude, what are you doing? This is the first call made on the phone! We actually got it all working!" <laughs> So, instead of being like this Alexander Graham Bell like, "Watson," whatever, I sent it to voicemail.

<laughter>

Grignon: So, the very first phone call went to voicemail.

<laughter>

Weber: Says something about the era as opposed to—

Grignon: Right?

Weber: —Bell's time.

<laughter>

Hsu: It's like it could've been—that's the first real voicemail, too.

Grignon: Yeah, well, it could've been something very prophetic, but it wasn't. It was just—

<laughter>

Grignon: —I don't want to talk to you. Go to Voicemail. <laughs>

Hsu: So, you were part of Fadell's part of the team and that was in Mariani One.

Grignon: Uh-huh.

Hsu: And Scott's part of the team was in IL2 on the second floor?

Grignon: Yep, yep. And obviously Apple now I think is very different. I mean most of the campus is in some way doing phone development or App Store development. But they had taken over an entire wing of Building 2, the entire half of the building. And it started off with just the doors, but they brought food in every night, so this was part of that culture which started that have and have nots, right? So, every night because the team was working nonstop, they would order whatever, some delivery service, Meals on Wheels, whatever, something like that. And people would see all of this food being carted in and they thought it was just some luxury lifestyle happening behind these frosted over glass doors. And the funny thing is is when you actually went back there, it stank because the janitor people weren't allowed to clean regularly, so all these food cartons, all gross food were piling up, 'cause they could come like once a week or whatever it was or twice—not every day.

<laughter>

Grignon: And I think people saw it as this pedestal and you go back and it's just a bunch of flies swarming around a garbage pit. But that's—

<laughter>

Grignon: —that was kind of that little area. <laughs>

Hsu: Wow. <laughs> I think I was the floor below it. I was on first floor.

Grignon: You were in first floor on the other half of that building.

Hsu: Yeah.

<laughter>

Grignon: Uh-huh.

Weber: How important was the development of wireless data in that period?

Grignon: Well, so I think iPhone surfaced the need for faster wireless data, because I mean if you remember that was like 2G. I mean it's the really slow Internet. It's one step above bare minimum. It was super slow. It's funny because I'm a T-Mobile subscriber and I travel internationally a lot, and T-Mobile infamously gives you data around the world—

Weber: I know.

Grignon: —but it's only 2G. And it's unlimited 2G data and you can spend extra for normal high speed data. But every time I go and travel overseas, I'm reminded this is what the very first phone data was, right? This was all we knew but we loved it because it was anything, right? And it's funny, when we look at—we've just become so accustomed to livestreaming video. We can do videoconferencing, FaceTime. We can do all sorts of stuff off of 4G or LTE network, but we've gotten addicted to it. And when you go overseas and all of a sudden they throttle you artificially to 2G, you're like, oh, man, this kind of sucks. <laughs> I guess I can use it to get maps on Google or find out when the subway is coming, but that's all it's really good for and that's all we had back in those days.

Weber: And was any of the negotiation with AT&T about trying to get higher speed?

Grignon: Well, so they all had a roadmap, yeah, and so, there wasn't much of a negotiation so much as visibility into the roadmap that they had already had, right, so there was a 3G deployment plan. We saw how over the next five years data rollout was gonna happen, and it's actually come pretty much on spec as they predicted it. When we start to look at 5G and you go to Mobile World Congress in Barcelona you start to see, wow, actually there shouldn't be much of a difference between my wired home connection and what I get on my mobile device. And I have no doubt that that's gonna be the reality soon.

Weber: Because Jeff Hawkins of Palm said that he thought the whole idea of a mobile web browser wasn't really viable much before that because there just wasn't the infrastructure. They tried with the Palm 7 and stuff. But I mean the iPhone came at the moment when the networks were just—does that—

Grignon: There wasn't, as I recall, much of a negotiation around the importance of higher speed data. We knew it was coming, but it wasn't a lynchpin, "Well, if we don't have it we're not gonna have a successful product on our hands," 'cause there wasn't ever an idea to stream iTunes music. You know, it was supposed to be resident. It was supposed to be local. The browser, you get what you get. I mean as that became more ingrained in our lifestyle, then we started to yearn for higher speed networking. And honestly I think that's one area where Android device manufacturers had some early successes over iPhone because you were locked in at whatever they had and the hardware capabilities at the time. And when you saw your friends with Android, even with a crappier user experience but with a much faster data pipe, all of a sudden that became a thing, right? Luckily, we've had a bit of a lull. Well, iPhone—yeah, luckily, they've had a bit of a lull and they're up to date now.

Weber: But then in the earliest version, the super high-resolution screen obviously is for the UI, but if not browsing what was seen as the purpose, photos?

Grignon: Photos have been a part of that initial experience since day zero, and it even went so far as pinch and zoom. Like, we wanted to show off multi-touch. And that was kind of the killer, ooh, ahh, sexy feature, right, when you could show somebody a thing and, <makes sound> and everybody just went gaga for such a simple little trick, but it was mostly around photos. And you see every iPhone announcement since usually has some play to the photography angle, and I think that's for a reason. I mean I think we've eliminated the need to carry DSLRs, and people are now—and today at Facebook F8, Mark Zuckerberg got up and said, "We're adding cameras to every application we've got, for anticipation of augmented reality." You know, it starts with Pokémon, which is a fun little game, but it quickly turns into something—look where we're seeing mass adoption. And all of this was a pipe dream 10 years ago, we're talking about. Did we have the ability to overlay live graphics? Well, kind of. We could put a button. <laughs> We had no idea that you could open your mouth and rainbow vomit would come out <laughs> on your Snap. But I don't think anybody foresaw that first as a real use case, but second, that was the stuff of computer SIGGRAPH pipe dreams.

<laughter>

Grignon: Right?

Hsu: Yeah. So, you mentioned earlier that Steve had to be convinced to do the phone.

Grignon: Yeah, because his very first foray, his version of dipping a toe in the water was the Motorola phone.

Hsu: The ROKR?

Grignon: The ROKR. It was an ugly device. It worked with iTunes but it really was kind of a disaster of a product. And Steve I think had convinced himself to not go into the phone business for a lot of reasons, regulatory issues, carrier problems, but then this tepid rollout of the ROKR, and he's like, "I don't know that there's a demand," so he had to be convinced, and they worked hard. I mean those guys, Fadell, Schiller, they all worked really hard to convince him, smartly so, obviously, but...

Hsu: Wow. I wanted to go back a little bit and ask about Dashboard again.

Grignon: Yeah.

Hsu: So, I remember when Dashboard came out as part of Mac OS X, there was a lot of controversy in the developer community because it was seen as a copy of a third-party product called Konfabulator.

Grignon: Konfabulator, yep. <laughs>

Hsu: What was your view of that?

Grignon: I mean it was silly. I mean it's not—of course we didn't. We never did. We know Arlo, who was the guy who wrote Konfabulator. We'd seen it, but this was—it's entirely plausible and it happens all the time where multiple people come up with the same idea at the same time. We executed I think ours differently and ours used stock WebKit, but it wasn't like, ooh, hey, there's this cool thing. No, we've got designs that stretch well before even Konfabulator existed—as Dashboard as a concept, and then when we built it it was a, this is such a natural technology for this. We had started to really push the boundaries of what CSS—you know, web technologies that we kind of take for granted today, but WebKit and any browser at the time couldn't do, including Konfabulator, so they ejected out on some of their core technologies that they used, because they had their own custom scripting engine that was web-like but not really JavaScript. And we had gone all in, but we had the luxury of controlling the framework that allowed that all to happen, right? I mean when you've got the team right down the hallway that builds the thing you're building your whole technology on, yeah, hey, great. And, you know, we had a lot of bugs in WebKit and there's a lot of extensions built for WebKit that were drawn from Dashboard but then made it as part of the standard, and so, we started to see some interesting things on that. So, was it the same? Yeah, you could look at it as a consumer and, whatever, but it wasn't.

Hsu: Hmm. How did you end up on the Dashboard, from going from iChat to Dashboard?

Grignon: I was just looking for something different, and iChat, it was iChat AV and I was done. You know, one of the things I've realized over the course of my career is I'm a starter. I'm not a maintainer. And there are people who excel at taking an existing thing and iterating and making it better. I excel at

taking nothing and making something. It might not be right and it takes a lot of effort, and there's a—sometimes being the first mover isn't always the best. You know, Steve—an infamous saying back at Apple was the pioneers get the arrows, right? <laughs> Kind of a reference to early America, and I think that's true, right? We've seen a lot of times where somebody was first to market, they failed spectacularly, but then somebody else came and did it right, and I think iPod is a great example of that. I mean how many MP3 players were out there? iPod wasn't revolutionary insofar as it was the first MP3 player. It was the one that did it right. And I think Apple is really good at waiting to see how a market matures. In the instances where they've kind of gone first, it hasn't always worked out, but I mean the iPhone wasn't the first [smart]phone, clearly. <laughs> We were just the ones I think that did it right. And insofar as when we introduced iPhone, Android was on the dock to be announced, and very famously Andy Rubin scrapped all the plans for Android after he saw the iPhone announcement and went back to the drawing board 'cause the bar was raised. So, now we look fast forward to 10 years now into it. Android has much higher adoption numbers just because it's free or because people like the hardware, whatever, but I wouldn't say Apple is getting their lunch eaten from them by any stretch, but Android has certainly become a formidable force. If I was Apple I'd be watching that.

Weber: Speaking which, the Google Maps was obviously a key part of the initial launch. I wanted to ask you about that.

Grignon: That was—so Google Maps, <laughs> for me it was that moment where I realized that we actually had a really cool product on our hands. And if you think back 10 years ago, I didn't have a real time GP[S]—I didn't have a Garmin or anything like that, and the way that we got around back then was you'd go to some website and you'd—not Google or whatever, whatever your thing was—who was it, tele—

Weber: MapQuest.

Grignon: Yeah, there was a bunch of them where you'd get a printout of the directions to your buddy's house or whatever it was, and you'd take that with you, and that's how you navigated. And I remember I was going someplace—this was before the phone was announced but I had UI privileges and all that stuff. And I was at a stoplight and I had no idea where I was going. I was like, oh, shit, what am I gonna do? And I remember I had my prototype, so I pulled out my prototype at the stoplight and I kind of got the directions from there. It was Yahoo's thing, whatever Yahoo's website was, yeah. And I looked up the directions to my friend's house or wherever I was going, and I was like, oh, my god. This is awesome. Even at its lousy 2G speeds, I could get the printed list of the turn by turn on where I was going. I was like, oh, this is great. <laughs> Now of course we've got real-time everything and Waze and Apple Maps and other <laughs>—

Weber: But that was in agreement with Google when it was still—

Grignon: Yeah, so when Apple and Google—

Weber: —that was a—

Grignon: —were still playing nicely together. And of course they very famously parted ways. There was that idea that Google was sniffing in on Apple's plans. I don't know how much truth there is. I wasn't there for that, so I don't know.

Weber: But when it was still good terms, did you see that as the beginning of something with them? Where would that lead, 'cause that's almost like an app but not really.

Grignon: Yeah, I mean—

Weber: It was almost like a permitted app or something.

Grignon: Yeah, I don't really remember that being as much of a thing as some of the other apps. I mean I remember being personally most excited about taking pictures, making phone calls, which really isn't all that exciting, but browsing the web. It wasn't necessarily maps so much, even though that was the first killer use case for me. And Google basically were the ones that wrapped their web experience around an app, right, so that was kind of the first developer story that they had, which was what paved the way for the actual WebKit SDK, right?

Weber: So, yeah, it did play a—

Grignon: It played a part, I think a pretty significant part from the developer's story. But Google did it the smart way, which was just wrap <laughs> their existing web interface into a mobile web interface, and there was just a shell.

Hsu: And that was the story for third-part developers as well, right—

Grignon: Uh-huh.

Hsu: —that Steve was saying just wrap web apps into—

Grignon: Yep. Well, and you saw, they were always a little off, though. Remember using those early web apps, and it was sometimes the—they didn't do it just right, so native apps were like a nice list scroll, kind

of had that bounce back and they had that feel, and we started—I think people's expectation of UX, initially it was kind of sloppy. They didn't know. You kind of liked your phone. It had a really nice look and feel. But then after like six months to a year people really started to demand a lot more out of that native look and feel, and it became very obvious when you were using a web app versus a native app. And when we look at—I think today even there's this concept in computer science where Moore's Law, you know, we have this huge doubling of processing power, and when we look at user experiences, I think those are the same kind of things. I mean imagine—not imagine. Look at any kind of new product release today. Look at how much ho-hum, lackluster, "Hmm, it's only got a 4K screen on it. I guess that's okay. It's got this camera," and this, that, and the other, and people are like, "I guess it's competitive with whatever." If you were to have <laughs> that kind of a product 10 years ago people would just be, their brains would melt. <laughs> "You can do what?" <laughs> And now we've just gotten used to not only UX but hardware improvements. I mean it's like, "Hmm, you know, the frame rate on it, it's not 120 frames a second. I can see the slightest glitch ever <laughs> when I'm doing whatever." You know, it's like, god, how entitled—or not entitled. How high is our expectations of what these things can do, not just from a software UX and UI, but all of the entire ecosystem. We're expecting full-on supercomputers in our pockets, and if they're not there then we lose it. <laughs> How crazy is that?

Hsu: On your LinkedIn page you've also listed that you validated the iPhone's custom CPU and you wrote the iPhone manufacturing line software?

Grignon: Yeah, there's software. I mean when you go—and it's not the sexiest thing. It's just when you go and make a new chip, like somebody gives you a piece of silicon, you put it in a test harness, and you can't just download a CPU tester 2000 app. You can't even use any of the basic libraries that you code with. There's no standard lib. There's no anything. And so, you have to write software that starts with the most basic of operations. It takes somebody who can code an environment where literally nothing exists. And so, you—'cause you're testing, can it add? Does it have fingers and toes, right? You have to start with the basics, 'cause if it can't add it can't do much more, right, so you have to write—you can't just bring in some library that does adding 'cause you have to check precision. You've gotta do, I need to move a certain pattern from this chunk of memory to that chunk of memory. Did the pattern move okay or has it gotten garbled in the transfer? Is it there at all? Did it crash? So, there's stuff that at its most primitive level it's bootstrapping a brand new piece of metal. And somebody has to do that, and we as software engineers are operating at a level 10,000 feet up. And somebody, whenever you're building a new thing has to start with zero feet. <laughs> Can it add? Can I move memory? Do I have a timer? The most basic, fundamental things. And you'd be surprised at the things you catch, and we should have caught things in that UART that went sideways. Like, we wrote a bunch of unit tests that test, can this thing move data? Yes, it can move data. We didn't check all of the conditions, and so, that was where we got bit on that.

Hsu: And why was it that you were the one that ended up—

Grignon: My teams were. I mean I didn't—

Hsu: Oh, okay.

Grignon: No, I'm a manager. This thing turned to tater tots a long time ago.

Hsu: <laughs>

Grignon: You know, my <laughs>—

Hsu: So, along with the radio, the telephony stuff, you were also doing these other lower level—

Grignon: Yeah, because when you produce anything, a surprising amount of software goes—even if you're just building a piece of hardware, right, our chips are built with software. Manufacturing lines are surprisingly a large amount of custom software at each step. As much as we've made of this person's job is to put this screw in this thing, you have to have a diagnostics level at every single point of the assembly line. Where does that come from? Well, somebody's got to write the code for it, and that's what we did.

Hsu: And so, your group just had to write whatever software to support the rest of the—

Grignon: Yeah, Apple is nice in that we knew how to build things, right? iPods taught Apple a tremendous amount of how a manufacturing line needs to be structured, not only for yield but just in problems that can happen along the way. You'd be surprised at, you know, unless you have that eye for detail the amount of crazy stuff that can happen. Random example was during iSight, the camera, we had—camera was a tiny tube about, what, five inches long, whatever, and the packaging was unique in that it was a nice, clear, acrylic case that it was put in, and Apple paid a lot of attention to that detail. And one of the things we found was certain stores across the United States were reporting when customers were opening their iSight cameras, the tubes were popped open, and Apple wanted a very clear—they wanted that kind of snap of the case open. They wanted to have that—you gotta, <makes sound> it kind of falls out of the tube. There's a very visceral connected—Apple pays attention a lot to those kinds of details. And so, when people would open their cellophane wrapping their lid was popped off and they're like what the hell is going on? And we went through and a product engineer had to go to the manufacturing line. They validated [that] every single camera that came off was sealed. And then somebody got on an airplane and they watched as the cameras were flying around the world, and what happened was in air, pressure differences, every single camera had the lid pop off in flight. And then when they'd land, <makes sound> it got sucked back in.

<laughter>

Grignon: So, they were all being sealed almost accidentally. However, when you land in a city like Denver, you don't have that pressure difference to suck it back, so a large majority of them were open. And so, the answer was very simple. It's a really small pressure hole that gets drilled at the bottom of the case, but somebody <laughs> had to go through like that, you know, like, huh, all right. Those are the kinds of details that Apple is really good at. And just like a small pressure release hole in a container, what you learn after you make 30 million iPods, you learn that unless you do this particular check at this stage in the assembly process, lots of bad things can happen. And so, we've just got to add a really simple test to just, is everything good or this particular check is good. You know, it was kind of a line at Apple, which was, building one of anything is really easy. Building a million of them is really hard. Building 100 million of them is even harder, because what you start to see from a quality perspective is the things or issues or bugs that appear in a small QA test, you know, five percent of the people report whatever, this crashes when you do that. Like, great. So, you can choose to ignore that. Or five percent in a population of 100 people <laughs> turns into a really large number in a billion. <laughs> And so, when you have—you know, a lot of companies would love to have these kinds of problems where, "Hey, we've got 400,000 people reporting this." All of a sudden that seems like a really big number, right? "Two million people are reporting that." You know, Apple's bug database is filled with huge, huge—'cause when you create a product at scale, small problems become very big ones in a hurry.

Hsu: And so, the software for the line was to sort of check those kinds of things?

Grignon: Yeah, because you want to make sure—I mean it's a really expensive proposition to have a return, and anybody who's gone through manufacturing hardware, having to deal with AppleCare, AppleCare is—or any care, whatever your thing is, right, that's an expensive—having a customer show up at a store, to the Genius Bar, is a very expensive deal. And Apple had to change the way they sold products because the bottleneck for selling iPods at the height of their popularity was their inability to move people through the store fast enough 'cause they couldn't accept credit cards fast enough, so that's how they invented the new way of coming to you with the little sideswiper. You don't sit at a register. That used to be the way it was done, but that was a bottleneck. So, when you have that volume of people, now imagine some percentage of them show back up and it's like, "Hey, I got a problem. My battery's not doing whatever." All of a sudden that's a huge problem. And so, self-diagnostics only get you so far.

Hsu: Earlier you mentioned that you're no longer allowed back at Apple. <laughs> Is that because of the circumstances of your leaving?

Grignon: Yep, yep.

Hsu: Okay. <laughs>

Grignon: Yeah, you know, <laughs> you can burn some bridges, and some people say, "Oh, everyone forgets over time." Well, sometimes not. <laughs>

Hsu: But I mean it's a new regime now, right, so...

Grignon: You'd think that. <laughs> But on the other hand, you know, you could also say that I don't want to go back to Apple. Two times is enough, and there's a lot of companies out there that are doing some really interesting things. Apple I think is in a different situation than it was kind of even a few years ago. As we start to see consumer patterns change you have to be able to adjust to that. AI is a huge thing, AR, VR, ML—machine learning, and who's leading the charge on that? Well, you can't say Google anymore. They're leaving Google, too, and Facebook. Where are they going? <laughs> Startups. They're going to Snap. They're going to a really diverse group, and it's not the case anymore where you've got a line of people out the door. Your top talent isn't going to the big brands anymore, and I think that's an interesting kind of shift in dynamic. When I kind of remember back to my own start, I was super proud to be part of Apple's ATG, and I don't know of a company that I could say that for right now, like, ooh, I can't wait to be part of XYZ Organization, right, because I think things have just changed. Brands have—I think it's different to be a consumer of a brand of Apple than it is to be an employee. And it's funny because you were there. When you go and work at a company like Apple, when they're monstrously—like in the height of the iPod, the number—as soon as somebody, "Oh, you work at Apple. Oh, you work on the iPod. Hey, I got this problem with my AirPort. Can you..." You're like, "I don't know anything about it. I don't know anything about your AirPort. <laughs> I barely know enough about the iPod."

<laughter>

Grignon: But now it doesn't have that same caché that it used to have. I think now it's, "I work at Space X." "Oh, you're launching things. That's cool." "I work at Tesla." "Oh, that's cool. You make self-driving cars." I think now it's a different world than it was way back when.

Hsu: Could you talk a little bit about what ended up happening with WebOS and with Palm?

Grignon: Yeah, it was unfortunate. While we saw I think a fairly—the product in its own right I think was a great product, maybe mistimed for the market. I think we were a little late, and I don't know that there was much we could've done to accelerate that. And we also did some wrong things and it was buggy. It had a weird advertising campaign. We had all sorts of weird things happen. But ultimately we saw an exit to HP, and HP wanted the WebOS platform. They didn't want the phones so much as they wanted the software. And initially I think people were skeptical of HP, right, 'cause it's a big company. HP is you and, well, at the time, 330,000 of your closest friends. And Palm, we were 900 and change, right? HR at HP at the time was 2,000 people—

Weber: <laughs>

Hsu: Phew!

Grignon: —so we were doubled up in HR people alone. As a VP at HP I was an elite member of 5,000 other VPs, 10,000 Directors. I mean it's just a company at scale, and you start to wonder how you actually accomplish things, especially when you're coming from a small, nimble—900 is small. When I started my own company it was six, right? <laughs> There was no department to handle anything. <laughs> And so, I think the HP acquisition for Palm was good because it did bring some resources we didn't have to the table. You know, they wanted to put WebOS everywhere, in printers, on computers. They wanted—it would've been—in addition to our own tablets and phones they wanted it everywhere. And, you know, unfortunately they had a change of heart. Well, they had a change of management, so their CEO, Mark Hurd, got left and then they brought in Leo Apotheker, who really didn't like us as Palm, as WebOS, And so, he set some milestones that weren't really attainable, and once we didn't attain those milestones then he killed the program. Ultimately WebOS saw a home at LG. And so, they make TVs now that have WebOS in them. They're smart TVs. It's kind of fun. They sold more TVs with WebOS than we ever sold phones and tablets.

<laughter>

Grignon: So, it's nice to see it actually still live on, but...

<laughter>

Weber: Can you describe when you were developing it, how did you see it as being different from iPhone at the time? What was the competitive edge?

Grignon: Well, we saw iPhone copy a lot of the features from WebOS. This idea that multiple—

Weber: But when you were first doing WebOS—

Grignon: Oh, yeah.

Weber: —just describe what your vision was of what it would be like.

Grignon: I think the most compelling vision for me personally was the card model. This idea—one of the things we saw, using iPhones, was that mode switch, the modality change from one app to another, and there was oftentimes where you'd get an email with a phone number in it and you'd want to effectively copy and paste it into an SMS message or whatever, and that was just a cumbersome way of interacting. And the model with the cards allowed you to very quickly go up into this nice card view, swipe over. Everything was live, so you could be playing a video and when it shrunk down into the cards it was still live and running and then you could switch over. And so, for me that fundamental user interaction change was one of the biggest deals. The other thing that we kind of touted was the keyboard, and the keyboard for a lot of people—a lot of people still like their Blackberries, not because the software is any good but because it's got a tactile thing that you can push. Autocorrect is a thing for a reason, <laughs> as hard—as we were talking about earlier about how autocorrect is very difficult to solve from a machine, problem, some people don't want that. They just work better—and I found myself on my various WebOS devices, I could sit there and drive and not look at the keyboard and just with keys I could do the thing you're not supposed to do, which is text and drive. But forcing you to look at a screen, you kind of have to—and lots of people have tried things like Swype and other different ways, but I think actual real keyboard still has a place, so I don't know. But when we were starting it, we just saw a—we always knew getting developers onboard was gonna be a challenge, and so, our lack of apps was gonna hurt us. We knew that, and I don't know that we ever got ahead of that. I do think the user interaction model was way better than anything we had on iPhone at the time, and so,—

Weber: And did you look at all to the old Palm OS?

Grignon: Mm-hmm. The old Palm OS had some really good features to it. I think one of the things that Palm always did well was organization and from that kind of business mindset of calendars and mail. That was never iPhone or Apple's strong suit, and now the fact that we came out of the gate with Exchange support, and those were things that iPhone ultimately caught up to, but we prioritized a different customer class and I think it showed. I mean Palm always had—they knew how to do calendars. They knew how to do contacts.

Weber: PIM-type things.

Grignon: PIM stuff. And Apple was very much geared towards the consumer, and that certainly touched those aspects but not necessarily the best at it. You know, the ability—one of the things I always liked and I took for granted until I went back to my iPhone after we tanked Palm was this idea that we would seamlessly blend multiple contacts into one, right? So, we could be friends on Facebook. You've got a Twitter handle that we're friends on. I've got your phone number in my address book. All of those become one logical person. And nowadays Apple has caught up to that, but back in the day you had to enter in a contact with all of those things, and you couldn't just have all these different data sources coming together being blended into one, I want to contact Mark. I don't want to contact Mark at Facebook or this or that. You know, I want to reach you. And so, I think we did that, but, again, it goes back to PIM.

Weber: And the synchronization, did you also—

Grignon: Yep, we did all the sync—I mean keeping everything in sync was actually a huge amount of effort for us.

Weber: And by the way, this I never tried to find out, but who had the idea of doing synchronization through iTunes, which seems like one of the worst possible ideas ever?

Grignon: Well, that was borne out of the iPod, so that was a vestige from the iPod days. You know, let's just use—yeah. <laughs> Let's just use iTunes as the source—now what's funny is you've seen them migrate off of iTunes, right, when they introduced cloud syncing. Nowadays you don't have to have iTunes anymore really to use your iPhone. It helps in some cases, but...

Weber: But you can't do a hardware—

Grignon: Well, yeah, if you want to do a reset—

Weber: —only through the cloud, yeah.

Grignon: —or a restore?

Weber: No, no, but you can only do the cloud-based without iTunes.

Grignon: Yep, yep, I know. Hey, I'm with ya. <laughs>

Hsu: You mentioned you worked on a printer at HP. Is that because they wanted to put WebOS on it?

Grignon: Well, you know, on LinkedIn I put that on there as a funny thing because the amount of effort HP put towards printers, I mean it's clearly the cash cow for a lot of what they do, and they make a lot of money on printers. At some point I thought it was silly to even put WebOS onto a printer, but, you know, I got to learn about how inkjets worked. That was kind of cool.

<laughter>

Grignon: It just seemed like overkill. You know, it's one of those—I think it's a typical example of kind of big company thinking, right? Hey, I got this hammer.

<laughter>

Grignon: I'm looking for anything that is resembling a nail. And their hammer was printers, and what can we do with printers? Well, anything, right? I mean you can—they wanted to embed a tablet in a printer. Wow, that's expensive, <laughs> but they were willing to subsidize it because they thought in their world printing was the center of the house… Okay!

<laughter>

Grignon: So, yeah, again, when you're struggling I think to innovate you start to do some interesting things, so that's more of like a snarky—yes, we did put it on a printer and it's about as great as you think it was.

<laughter>

Hsu: So, you left HP in 2012—

Grignon: Yep.

Hsu: —and then you started Quake Labs?

Grignon: Started Quake Labs. So, I wanted the exact opposite of HP. So, instead of 330,000 people I wanted six. And, luckily, with the career I had raising money wasn't a problem, so a venture firm called True Ventures gave me my first million to me and Jeremy from my Iowa days and we started a company together. And the best part was—well, best and worst, we never had to pitch. We got the money with the idea that we could come up with something interesting, and we did, actually. It took us about six months to really figure out what the product we wanted to build was, and the product was we had both seen the stratification of Android versus iOS and within Android and iOS you have tablets, phones, TVs, whatever. If you were a software developer, good luck on making an experience that spanned all of them, right? It was hard enough to find—because even within Android on a phone you've got how many different screen sizes? You know, you've got different ways to input. You've got all the things. How are you supposed to pick a platform to go to market on? And what if you don't really want to create an app? What if you just want something kind of light like a photo book or whatever? And so, what we wanted to do was regardless of whether you were on Android or iOS, whether you were on a tablet, a TV, or a phone, we

wanted to allow regular people to create something that spanned all of them, so we did. We invented a new programming model that was approachable by nonprogrammers. Think of it as HyperCard for today. And I introduced it at LeWeb in Paris a few years ago, and it was really well received. And so, we got it out there. People loved the card model, so it was—when I was raising money I had, like every startup, a pitch deck, right? Here's this thing and this is how great we are. Here's our vision and here's my bank account. Just wire me the money.

Hsu: <laughs>

Grignon: Questions? Cool? No. So, I pitched probably 80 different venture capitalists, and getting in the door was not a problem. Getting a vision that they understood and kind of latched onto, completely my problem. And nothing—the story that I was telling didn't resonate with anybody. I signed on an advisor who kind of looked at what we were doing. He was like, "You're not telling a story. You got to tell a story." And so, we kind of reinvented ourselves and we're like, "Hey, our thing is programming with cards." So, what we did was we changed our pitch deck. I threw away the pitch deck on the PowerPoint. We printed out a deck of cards. And the very first time I told this story I went to a venture capitalist, it was Lerer Ventures, and I was like, "Let's build an application." And so, I started playing cards like it was like magic or something.

<laughter>

Grignon: You know what I mean? It's like, "Imagine this is the thing. This is how it all works. So, we have now just built an application with our stuff." And first time I did it he was in, and closing venture capitalists after that, we got Google Ventures, we had Lerer Ventures, Betaworks. It was led by Jerry Yang and Ash Patel from Yahoo. It's all about telling a story, and once we got that we started to get that momentum. And so, we built a product and we built a platform, and it was a platform of platforms, so it wasn't Android, it wasn't iOS. It did all of it. And after we burned through that second round of cash there's a decision you have to make. Do you want to raise money or sell the company, and ideally do both <laughs> but we were at the end, and so, we decided in order to really be successful with this platform we needed the pockets of somebody much bigger. And we could go raise 15 or 20 million or we could be part of Apple or part of whoever. And we got to the altar four different times as an acquisition and it fell apart four different times, but for the same reason, corporate restructuring. So, every single time—Google was interested and we went through the whole process. I mean it was great. Google interviews every company that they acquire, so they came up with a really cool little interview process for us. Nobody had to write code. We were all seasoned enough that that wasn't an issue. But I show up to my first interview and my champion is, the guy who was spearheading the acquisition—was my first interview—I'm like, "Hey," so it was just small chit-chat. "Hey, how's it going?" It's like, "Well, we just had a pretty major restructuring yesterday and I don't know my role at Google going forward." I was like, "Oh, great." It was Google+ and they just killed it the day before. It was like—but that's startup life, right? Things happen. And so, we ultimately decided instead of selling it to a company that wasn't really interested in taking care of the investors, we

had some offers on the table, and they were not at companies that we really wanted to be a part of, we just decided to fold it. And so, we killed it. And sometimes you just have to make that decision where you have a great thing. You've got some great IP. You've got a lot of promise, but then you just have to know when to call it. Hard lesson about being an entrepreneur, but when I talk to entrepreneurs, I invest and I counsel and I advise and I'm on boards, those are all the—obviously you don't want that outcome, but it happens to most of them. Better to have recognized the signs and how to move on than to beat the dead horse.

Hsu: <laughs> So, since then, 2015, you joined Siberia?

Grignon: Yeah.

Hsu: What do you do there?

Grignon: So, yeah, I joined Siberia. After I had folded Quake Labs I was trying to figure out what I wanted to do, and there was a lot of offers coming in from just big companies. And when you go from your own thing, your own small company, the idea of going back to a big company isn't appealing at first, and I didn't really want to hop right back into a multi-year commitment about thinking about the one thing every day. And Siberia, a friend of mine had started the company and he put out a tweet, "Hey, we're looking for a managing director for San Francisco," blah, blah, blah. It's an agency. It's a design and engineering firm, and—

Hsu: So, they're a consultancy.

Grignon: Yeah, it's a consultant agency. So, he put this tweet out and I just half-jokingly responded, "Hey, you looking for management overhead?"

<laughter>

Grignon: He was like, "Are you interested? Let's do this." And so, we met for breakfast the next morning, and what sold me on Siberia was, we work with some of the world's biggest brands and we don't talk about who they are because our success is their success and it's not important. At the end of the day to bring new business in, yeah, we'll tell under NDA, but we don't really talk about the brands. But we engage at the CTO, CEO, EVP level, and these brands that we work with, we get to see—they open the kimono. We're there to help them solve problems, and so, they don't hide things from us. You know, we're there to help them innovate and come up with some really great things. And so, what I was hoping to get out and which I have been getting out of Siberia is, what are the world's biggest brands thinking about? Not necessarily technology brands, either, so clothing, wearables, all sorts of things, what are they

worried about? And what's interesting has been they're worried about the same things that venture money is being poured into. So, imagine a clothing brand. They're looking at ways that they can use artificial intelligence to help them sell more clothing, but they do clothes. They don't do software. They don't do technology. Car rental business is the same thing. Imagine a big car agency. How can they use technology to make that process better and faster and easier? And so, again, they don't have armies of software engineers like Google or Facebook does. You know, they rent cars <laughs> and they have IT departments, but they don't necessarily have innovation departments or anything like that. And so, it's been really fun to engage with nontechnical brands that are trying to figure out ways to use technology, and it goes right back into machine learning, artificial intelligence. We're just now starting to see some interesting stuff around their ideas around virtual reality, not even augmented reality. You know, how would you have a better hotel rental experience if you could see the room or how to move around from the foyer to your room or to the dining room, whatever? So, they're interested in looking at ways to embrace cutting edge new things, but the best part is we do this for six months, a year, whatever, but we have like eight of those happening, <laughs> so you can flit between little things. So, it's the perfect thing for somebody that has an attention span of a gnat—

<laughter>

Grignon: —because it's all new. Going back to my career, right, I don't do well with innova—I'm sorry. I don't do well with iteration. I'm really great at those first, like, "Oh, you know what we could do? Let's get a team on it and let's start building this." And then, "Okay, that's done? Let's go do it again and let's do it again. Let's do it again," and that's not really traditionally the strength of big companies that want to, "All right, let's invent a new thing and then let's just bury it <laughs> for the next six years," so that's what's been a lot of fun about it.

Hsu: And is there anything you'd like to add, especially looking back on the last 10 years of the iPhone?

Grignon: You know, I'm glad iPhone turned out to be a success. You know, it was a very difficult product to birth, and I learned a lot, not only about myself, about how other people work. You know, it's the most extreme—anything. It does make other challenges that I've had—just not professionally—personally, it puts them all in perspective, right? Once you've been <laughs> put to the test, you've seen as bad as it can get, actually everything else is pretty easy. <laughs>

Hsu: I mean your personal life was pretty severely affected.

Grignon: Oh, yeah, yeah. What I say is I got a divorce because work is my mistress, right? One thing Steve was really good at was identifying people who will do anything it takes to ship a product, and that works for and against you, right? I'm your person to deliver a new cutting edge product. I will do that at the expense of literally anything else, my own health, my marriage, anything else, relationships. And

some of these products, and the iPhone is one of them, requires that, not just from me but lots of people, and that was kind of a pattern when you look at everybody in aggregate, all kind of the same. You can't kind of limp in and just say, "Hmm, yeah, I'm gonna go home now. It's eight o'clock and I've been here for..." No, it doesn't work that way. <laughs> You go home and then you work more. One of the jokes that we used to say was, the greatest thing about Fridays is it's two more working days until Monday.

<laughter>

Grignon: So, I think, yeah, I'm glad it was a success. I wouldn't take it back. I wouldn't do anything really differently. I would wish, I would hope that we as the consumer, the consumer population, figures out a better way to utilize technology in our lives more. I mean it's almost funny when you see people falling into fountains and things like that because they're so engrossed in this little screen. But I don't think that's the fault of an Apple or a Samsung or whoever is making the phone du jour. I think that's just what people do. So, I wouldn't say it's a fault. It's exposing the usage patterns of regular people. So, I would like to see us <laughs> embrace technology in a maybe more healthy way. <laughs> You know, this idea of, when Caltrans has to put on their signs, "don't text and drive," like that's <laughs> a new thing? That's a sad reminder that people can lose their lives over stuff like this. And so, I guess that's one of the things that I would probably not do differently but I would probably spend more energy towards addressing.

Hsu: Thank you.

Grignon: Sure.

END OF THE INTERVIEW

