Lecture 12 - Building for the Enterprise (Aaron Levie)

46 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=TFVDJRVQJDW&LIST=PL5Q_LEF6ZVKATY_CT1K7QFNF2TIDHCE-1&INDEX
https://www.youtube.com/watch?v=tFVDjrvQJdw&list=PL5q_lef6zVkaTY_cT1k7qFNF2TidHCe-1&index=12

SUMMARY

Aaron Levie, CEO and co-founder of Box, discusses the journey of building an enterprise software company, emphasizing the unique opportunities and challenges within the enterprise sector. He argues that the shift towards cloud computing, user-led IT models, and the need for innovative solutions present a favorable environment for startups in enterprise software.

- *Levie aims to convince the audience of the merits of building enterprise software over consumer-focused products.*
- *Box started in 2005, initially not targeting the enterprise market but pivoted due to the growing need for file sharing and storage solutions.*
- *The enterprise software landscape has changed significantly, with a shift towards cloud computing and user-driven technology adoption.*
- *There are substantial financial opportunities in enterprise IT, with $3.7 trillion spent annually compared to $170 billion in consumer apps and advertising.*
- *Startups should focus on spotting technology disruptions and starting small to find niche markets before scaling.*
- *Leveraging asymmetries in the market can help startups do what incumbents cannot, often due to economic or technical constraints.*
- *Listening to customer needs is crucial, but startups should distill these requests into innovative solutions rather than build exactly what customers ask for.*
- *The current environment is ripe for launching enterprise software companies, with significant potential for growth and innovation.*

Can we keep playing, wait,
okay good. Can we turn it up a little
bit, so it's more pumped up? >> That's loud. >> Okay, here we go, okay. >> Okay, so we gotta find
the beat and then we gotta clap to the
beat. >> Okay. All right.

Okay, that's pretty good
guys. All right, we're good. We're good. Thank you. Okay, please stop. Okay, stop the music. Thank you. Okay, cool.
Can you put on the presentation, thank you. >> That'll be the most
pumped up thing that ever happens in
enterprise software, so the rest is sort of downhill

from here. >> So, thank you for that
intro, that well-rehearsed intro,
thank you. >> Confirming the facts as
you're saying them, very good. So I'm Aaron Levie, CEO and
co-founder of Box.

Welcome to this edition of
how to build an enterprise software
company. That's my understanding.
This is the course that you're taking, right, is
that correct? >> No, okay, so. >> Here's my job today. My job is to try and convince you that everybody
else who speaks throughout this whole class is wrong
and that you actually wanna build an
enterprise software company. And so hopefully we'll be
able to kinda work through this, and
you'll have a good sense of why it's super cool to be in
the enterprise.

And why a lot of the
perceptions of going into the whole
consumer space, where it's, you know, so much fun and
everything. Why that's wrong and why you wanna do enterprise
software. Who wants to build an
enterprise software company? Good, all right. Thank you very much. Hopefully we'll do a vote at
the end and hopefully that will not have
shrunk. So that's really actually
the only goal I have at this point. So, we're gonna talk about
three things today.

The first is the quick
background and history of Box. Cuz when we started the
company, we didn't know that we wanted to do enterprise
software and so I wanna kind of get through a little bit
about why we decided to go after the enterprise and
what we do today. And then I'm gonna talk a
little bit about what are the major
kind of factors that have changed in enterprise
software that make it possible to do a start up
today in this category.

And then finally we'll talk

a little about some patterns that you
can kind of exploit and some ways to recognize and
ways to actually go build an enterprise software
company yourself and hopefully that'll be some
practical, useful advice. Just as a forewarning, my
voice, as I've been speaking a lot
the past couple days. So hopefully, I'll be able
to get to that third part of the advice and actually be
able to talk, so we'll see how we make it but I'm gonna
be downing a lot of water so I apologize for that.

So building for the
enterprise. This is what we're gonna
talk about. So just quick kinda high
level stats about Box. We've about 240,000
businesses that use the product. There are over 27 million
users that have brought Box into their organization and
used it as individuals and we're in about 99% of the
Fortune 500. Unfortunately that last 1%
is really just Microsoft. And they don't seem to wanna
buy from us. So. >> We have to work on that a
little bit.

But certainly, a decent
amount of scale, a lot of users that have
brought Box into enterprise environments,
these are some of the large organizations that
are using the product. So, a very wide range of
industries, everything from manufacturing and consumer
products to companies like General Electric. Stanford Healthcare actually
uses the product for collaboration within the
hospital environments and the research. But between healthcare,
media, manufacturing.

These are the sort of the
range of industries that we serve. So, the question's how did
we get here? Cuz we didn't really start
the company to try and go build an enterprise
software company even though that is how things
ended up happening. So when we started the
product, so we launched the company in
2005. We got the idea actually in

college back in 2004. And so, was anybody using
the internet back in 2004? Okay great.

So basically, I don't know
if millennials use the internet
back then. Sorry. Okay, no more age jokes. All right, here's the point. So, back in 2004, you might
remember that there wasn't a lot to do on the internet. It was actually kind of
boring right? This is before Facebook. This is certainly before
Snapchat. So not a lot of things that
you could go to, you couldn't send people 15
second messages and photos that disappeared, cuz you didn't even have a
phone to send them photos.

So in the internet in 2004,
there wasn't a lot going on. This was basically what the
internet looked like. Sort of a barren, deserted
landscape and this was the internet and
just to clarify, the happy camel's Google and
the sad camel's Yahoo! >> And that was effectively
the internet and this is accurate, this is
the internet in 2004. Yahoo has done a lot better
since then but, but back in the middle 2000s they were certainly trying
to find their way and Google was sort of taking
over the world but this was the entire, this
was the extent of the world.

So, what we noticed in 2004
in college was that it was, for some reason, and this
isn't so much the case today but back in 2004 it was like,
really hard to share files. And as simple of an idea as
that is now, you go back ten years and it was either
really expensive or it was really hard to move data through different
corporate networks. So I had an internship at
the time, and in the internship, most
of my job using data was actually just to copy
printed out pages and then put them in cabinets.

That's what you do as an intern if you're not a
computer science student. So I was really, really good
at copying paper. And unfortunately not a
skill that has really been useful today. But, it was really hard to

share files and then in classroom
environments you were constantly working with
lots of other people and it was also hard to share
files. I went to USC. And USC gave you 50
megabytes of online storage space in your email account.

50 megabytes. So you can basically store
about one file and then it would auto-delete
every six months. I don't know who was running
IT at the time but they certainly didn't buy
hard drives. So, there was just really,
really hard to store and share files. And so we said, well, why don't we make it really
easy to store and share files from anywhere? And so we got this idea for, at the time it was called
box.net.

And what we noticed was
there were a bunch of factors that had changed in
the software world. The first was that the cost
of storage was sort of dropping dramatically. So, in our business,
basically every year or two, you can double the
amount of data and storage that goes in to a
hard drive. So, what was uneconomical
two or three years ago, all of the sudden becomes
feasible because the cost of computing and the cost of
storage has dropped.

We had more powerful
browsers and networks so Firefox was just emerging,
people were using the much faster internet in both
their homes and in the classroom. And then people had more
locations that they wanted to store and share
information from. And so we had these sort of
three factors that all of a sudden were emerging. And I'm gonna sort of pull
these factors back later when I give some sort
of tactical advice but sort of the first point to
remember is, always look for these changing technology
factors.

Cuz any market that has a
significant change in the underlying, either raw
materials or enabling factors is an

environment that's about to change in a very significant way. And so we were very fortunate that basically the need for data in the cloud was growing in importance, and the cost and the feasibility of doing it was also improving rapidly. And so we decided to put together this really quick version of Box, we launched it as box.net, and the idea was, let's make it really easy to share files.

And it turned out that the idea clicked. We got angel funding from this guy named Mark Cuban. And this was before Shark Tank, but it was very similar. And so, we got this funding, and we decided that, okay. This is gonna be super exciting. We are going to drop out of college. We're gonna move to the Bay Area, and it's gonna be totally awesome. And when you drop out of college. Anybody drop out of college yet?

Okay, good, stay in school. So when you drop out of college, like everyone sort of pictures it, it's just gonna be, you know, incredible, like Bill Gates dropped out of college. It will be like Bill Gates, or Michael Dell dropped out of college and it'll be super exciting like Michael Dell, or Steve Jobs dropped out of college, so this is what people imagine. But nobody ever remembers that this guy dropped out of college also. >> And, so it's not really a guarantee obviously that it's gonna be successful.

But, it's what you do, and actually, funny, I don't even know if this guy dropped out of college. >> It just sort of looks like he had to. >> I apologize if anyone's related to him, it's just a funny picture on the internet. So. >> So basically we decided we would drop out of college. We moved up to first Berkley and then we moved to Palo Alto a little bit thereafter. And we decided that we were gonna open up the product for free.

We got hundreds of thousands
of people that would sign up for the product every single
month. So, we had a free version of
the product called, if you went to box.net, you
got a free one gigabyte of file storage space, which
was, again, quite a bit back in
2006 at this point. But we were getting so many
users and we were trying to figure out
what to do. And what we ran into was a
common problem that really any startup runs into but
was really pronounced by our business model, which
was that for consumers, we had built a very robust, a
very comprehensive product.

And for enterprises, we had actually built too
insignificant of a product. So, for consumers, what we
were running into was, we had all these features that
you could pay for but a lot of the consumers didn't need
all of those features. And for enterprises, we didn't really have
enough, security. And we didn't have enough
capabilities around how enterprises wanted to use
their data. So we basically had more
functionality than what a consumer wanted and less
than what enterprise wanted.

And so we found ourselves at
this juncture. And we found ourselves
basically in this period where it's very
difficult to figure out what we wanted to go do with the
business, so we had to make a choice,
right? And so we were at this path
where basically, we're at a juncture where we
had to choose which path do go
down. And this was back in early,
mid 2006, up to late 2006, so I was, I think, 23 at the
time.

The co-founder was, was 22. Our founding team was even
younger. When we all dropped out of
college. And so back in 2006, 2007,
we imagined these two paths and the worlds were
very, very different. Like when you do a consumer

start up, it's basically just lots of
fun. And, you know? You have parties all the
time. And it's just super
exciting. And then the enterprise, you basically are battling
these just large incumbents. It's a fairly thankless, you
know?

Model because you know? People just generally hate
enterprise software. So that was sort of how we
imagined that the two paths was, we had to choose,
you know, one of these two worlds. And so we looked at that and we said okay, well consumer
looks really fun, enterprise looks really hard
and, and there's a lot of
competition. At the same time, in the
consumer space, you are always fighting this
issue of, how do you monetize? How do you, how do you actually get people
to pay for products?

And in the consumer space,
there's really only two sort of business models that you
can do. You can have people pay for
your application. Or you can provide
advertising on the application. And so just to give you a
little bit of perspective, these are today's numbers. Just to give you a little
bit of a sense of the scale difference, in the
consumer world, there's about $35 billion
spent on mobile apps every year,
pretty big number, right?

$35 billion that's, that's a lot of money being
spent on mobile apps today. For advertising, the global
digital advertising market is $135 billion. So, you know, basically
combined most consumer companies are
going after, if you're not doing e-commerce, about $170
billion of either sort of purchasing
power on applications or global advertising around
these kinds of services. So, big numbers, lot of
opportunity there, however, in enterprise, there's
actually $3.7 trillion spent on enterprise IT every
single year.

So these are the servers,
the infrastructure, the software, the
networking, the services. All of that stack of
technology equates to a few trillion dollars spent
every single year. And so what we realized was,
okay, there's a pretty wide delta
between these two markets. We are gonna be fighting
over trying to get consumers to pay a few
dollars a month. And Google and Microsoft and
Apple will probably try and make this product be free
over time. And there were sort of
rumors that Google Drive was coming out and these kind of
products that eventually happened were
going to come out.

But in the enterprise, it's
not so much about that they are trying to always save
money on IT. They're actually trying to
increase productivity. They're trying to get higher performance from
their businesses. So the value equation is
very different right? In consumer, we have a
limited amount of money that we want to actually make
sure that we conserve for as few things as possible
that we're gonna spend. In the enterprise, it's a little bit of a
different shift, where you're trying to think
about actually, what can I get out of
technology, and how much value is that for
me.

So, that was a really
important data point. However, the problem was
that enterprise software was really, really unsexy. Right?
So again, very competitive, very difficult to build a
business. It wasn't something that
really you should have shot out of bed in the
morning and said, okay, I'm super excited to build an
enterprise software company. And the reason for that was actually pretty
straightforward at the time. The way that you built
software was very slow. So you had to be very slow
because you couldn't break anything
for customers.

The sales process was very
slow because customers take a long time to purchase
technology. So I think everyone's sort
of used to this concept that usually when you try and
sell enterprise software to a company, it can take up to
a couple of years for them to actually just buy
the software. And then they can take a
couple more years for them to even implement the
technology in the first place, so a lot of
companies are around for a few years without having
their technology even used in the enterprise.

That felt like a huge
problem. And not like something we
wanted to be a part of. The technology itself is
fairly complex. So the user experience, I
don't know how many people have had to use most
enterprise software, but it's generally really
complicated. You find yourself asking why
in God's name did the designer try to put 47
buttons on the page. And you just can't even
understand it, and the reason is something
we'll get into in a second.

But basically, there's just
no love or care for the design and user
experience, the software's just really
complex. And then, finally, if that
wasn't bad enough, all of a sudden you had to think about how do actually
sell the software. And, for anybody who loves
the power of the internet. This notion of having a
sorta sales intermediary, to get to your customer, seemed
like, really unappealing. Like, you're gonna have to
go hire a bunch of people that, that are gonna be
everywhere in the country.

And they're gonna be the
only interface, you have to your customer. You're gonna have to hire
these guys named Chuck. And Chuck's gonna roll in
with a briefcase, and he's gonna just try and sell, lots of enterprise
software to the customer. Just so we're clear, this is
what Chuck looks like. And that was the sales

process that you, in the enterprise, that we at
least imagined in our heads. And I mean, Chuck looks like
a happy guy, but he's still an intermediary
to getting your software.

And we were saying, well, why can't we take the power
of the Internet and actually, get our technology
out to people directly? Why should we have to go
through this sales intermediary as we
scale the business? Now, I'll get into, in, in a
minute, why we were wrong about, the
sales process but this was the sort of fear
that we had at the time. And then, if that wasn't
hard enough we had investors in 2007 saying
basically, there's no way you're gonna
make it in the enterprise.

You basically are again, a founding team of, of early
20 year olds. You don't have anybody on
your team that has been in the enterprise. The Microsoft and Google and
all these things are, or not Google, but Microsoft,
EMC, Oracle, IBM. These kinds of companies are
gonna stomp on you and it's gonna be very, very
hard to actually achieve. And to be fair they were right on a number of areas,
right? So we were a very
inexperienced team.

This is this was a, in a stage where it is still
early in our careers. My co-founder, for instance, looked like he was
13 years old. So just not really, just to
be clear. This is what he looked like. So So it, it sorta made
sense. Right? This is, this is him as our
CFO at, I think this is him at 29. But it looked like we were
gonna go run off with the money and go to
Disneyland.

So I can totally appreciate
why they didn't think we would be able to pull it
off. I can't imagine giving him
money. So basically, we decided
okay well we still have to go do it, we have to just

give it our best shot. We're gonna take the sorta
scale, the consumer experience, the
DNA of our company. We're gonna see if we can
bring this, into the enterprise and then
we were very fortunate where actually we
had an investor who was equally early in his career
and made a bet on us.

Because sorta with the
belief that there was something fundamentally
changing about the enterprise that we would be
able to take advantage of. And so, we decided that if
we're gonna do the enterprise, if we're gonna
go after the enterprise. That we would have to play by a very different set
of rules. So, what about the
complexity of software can change in this
new era? What about the fact that the
sales process is very slow, can change in this new area,
era?

How do we move and go
directly to the user and to the customer. As opposed to having this
very, sorta indirect process of getting our technology
out there? How do we build, in design
for that user as opposed to, just for the sale and sorta
for the RFP process that a customer is going to go
through. And so we looked at all of
the factors, that were true of the
enterprise. And we said, we're gonna do,
maybe not in all cases, the opposite, but we're
gonna find what is changed about the technology world.

Where we can take advantage
of that shift and build a, a better and a newer
enterprise software company. And so, that was the decision that
we embarked on. This is the path that we
embarked on about eight years ago. And that is why, we've been
focusing on the enterprise. And so today, again, we have
about 240,000 businesses that use the product. And the reason is, we
architected the business model, we architected the
software. We architected the solution,
to work in sorta one specific

version of the world.

And it turned out that that
version ended up being, the sort of, the one that
happened. And I'll go into a little
bit about what has changed about
the world. That we sorta built out
company around. And what, I would highly
recommend you, if your building an
enterprise software company, make sure to orient to your
technology and solution around. So, that was, sort of, why
we made the decision, and kind of how we started
to take on the problem.

Now I'd talk, like to talk a
little bit about. What has changed about the
enterprise, that makes it sort of so
possible to enter it today, better and more seamlessly
than ever before. So, everything about the
enterprise and then by definition the software
that an enterprise uses, has changed just in the past
five years. This is probably the most
magical time. If there could have ever
been a magical time to build an enterprise software
company now is absolutely that time.

Just in terms of how much
change is going on within organizations. So let's just go through a
couple of these things. So, the first is that most
application categories are moving to the cloud. And the big difference is
this idea that if you were going to start a customer
relationship management software company, or a business intelligence
software company. Or even a content management
company ten, 15 years ago, you basically had to go
have your technology be implemented, in every single
customer location.

No matter how many customers
you sold to, no matter what regions
they're in, every single customer had to put that in
their data center. And that was the flaw with

on-premise computing, was you were repeating all
of this work. You were, creating so much
redundancy. And it was, it was then slowing down the entire
process of delivering and building software for the
enterprise. And then all of the sudden
the cloud came around. Things like Salesforce.com,
things like Amazon Web Services that
basically said.

Why is it that, that every
customer, that wants to just implement
a couple servers, has to go procure the
servers. Put them in their data
center put all the security, all the networking around
those servers. And then, and then, six months later they
go live and then a developer can use
them in the organization, or same thing with an
application. Amazon said why does that
make any sense today? When we could just, put
together thousands, tens of thousands, hundreds
of thousands of servers and make them available on
demand.

And you just use what you
want when you need that. And that was, that's
obviously a definition of cloud computing. What is happening, though,
is finally CIOs in large enterprises are taking
advantage of this. So it's, it seems obvious
for everyone in this room
because you would never build, a company by
buying your own servers. You would start it on
Amazon, or, or Google or Azure or whatever. But to an enterprise, there are decades, literally
decades of investment in infrastructure, that now has
to move to the cloud.

And so that's obviously a
massive shift that is finally happening. We are moving to a world of
cheaper, low cost, on demand computing, from a world of
expensive computing. The benefit to building a
start up, is that customers don't have
the same kinda friction. When they're gonna go adopt
new technology, right? As soon as the computing, is

so much cheaper, it becomes easier to adopt
new solutions. Which means, that their
barrier for having a conversation. Their barrier for
introducing you then, into their enterprise, is a
lot lower, which creates a massive
opportunity for start-ups.

We're going from a world of
customized software, to really standardized
platforms. So it used to be that you
had to build custom integration's
and all of the custom experiences on
top of the software itself. And now, customers are
realizing that actually they want, open platforms and
they can customize at the layer above the, the
product. It used to be that when you
were building enterprise software company you could
only really sell to, the top five or 10,000
companies in the world.

Because only those
companies, had the where with all, the
talent, the infrastructure and the
budget to deploy your technology into the
enterprise. Today, literally a two
person company can sign up for box, as well as we work
with General Electric which has over 300,000 employees. So, the fact that you can
now serve a small business anywhere in the
world. As well as some of the
largest companies on the planet, mean that there's much larger markets
that you can go after.

Which makes it, obviously,
even a better economic proposition to go after the
enterprise. The platforms themselves are
becoming more global. So our customers were
international literally within the first couple of
weeks, of starting the company. It would have been, usually
if you had done enterprise software the traditional way
that would have taken years, to actually be able to go
international. And then finally and probably the most profound
shift of all. Is that because of mobile

devices, you know, iPhone's, iPad's, tablets
Android devices, the IT model of the enterprise has
become a lot more user led.

And, that's fundamentally
important because, in an IT led world,
incumbents generally win. Because, they have the
existing relationship with the IT
organization, with the CIO, with the spending power
within that company. In a user led model, users
are bringing in, their own technology. They're bringing it in on
the sales team, they're bringing it in, in
the marketing team, they're bringing it in in
finance. And you can build software
then around the user. Which means that they can
bring the technology in.

Then you can sell to the
enterprise when they wanna have better control, better security, better
scalability. So you still have the same
business model, as a traditional enterprise
software company. But the way to get into the
company is now, through the end user. Okay, so those are
qualitative factor changes. Just a couple quantitative
factor changes. There's over, nearly 2 billion smart
phones on the planet. That changes, every single
IT model, on the planet, because it used to be, ten
years ago, if you were managing technology for
an enterprise.

You just had to manage the
computers and the network that were inside
of your building. But now with billions of
mobile phones, you fundamentally have to be
able to manage the network and the
computing. That is gonna be anywhere at
anytime on a network. And that creates massive
opportunity for software companies, because no incumbent has
built the technology stack. Sorry for my voice breaking
again, so no incumbent has built a
technology stack. That powers this next
generation of work, and how enterprises are using

their data.

So that creates a massive
start up opportunity. There's nearly 3 billion
people online. What that means is that
every single enterprise is equally changing how they're
gonna get their own products to their customers. Which means, that every
industry changes. And so, there are really
only two times, there are only two moments
of opportunity. Where a technology
revolution will happen in the enterprise. The first is if the raw
materials change. So, cost of computing goes
down and then it's better to centralize, and then let
people use it on demand.

The second major change is
that the very customers that these enterprises have to go
after, need new experiences of working with
those enterprises' products. Let me give you an example. If, you know, when you go
out off campus, you probably use something like
Uber or Lyft, as an example. The if you were in the
transportation business. If you're in the shipping
business, if you're in the logistic
business, Uber represents a massive
change to your industry. So you can't just let Uber
exist and do its own thing without
understanding.

What are the implications of
Uber? What are the implications of
InstaCart? What are the implications of
Lyft to my business model? And so, in a world where, enterprises are dealing with
that kind of change. They're gonna need new
technology, to help them evolve their
business models, and how they adapt to this
disruption. And so in every industry,
and this is why it's an amazing
time to even build vertical software companies for
industries. Right now, every single
industry is going through a major business model and technology oriented
disruption.

Which means that they're
going to need technology from startups to
help them work through this. I'll give you a couple of
examples. So, in the retail industry. There's this notion of sort
of omni-channel or multi-channel commerce. You're gonna shop online, you're gonna shop on your
phone, you're gonna shop in a
store, and you want things to be
delivered to you as well. So most of the incumbent
technology in the retail industry, doesn't power
multi-channel commerce. No, nobody is prepared for what does it mean when
consumers wanna go actually buy goods at anytime from
anywhere.

With better information and
better intelligence. So every retailer in the
world, is gonna need a new technology stack to power
their retail experiences. In the health care space,
every single health care institution is trying to
find ways of building a more personalized experiences,
more predictive experiences. They wanna have medicine be
adapted to the individual. As the business model of health care changes from
being about charging for the surgery, charging for
the checkup. And instead where the
customer pays for wellness and staying
healthy.

Then all the sudden every
healthcare institution, needs better technology to
deliver health care experiences. They're gonna wanna deliver
telemedicine. They're gonna wanna deliver
healthcare in more regional locations as
opposed to just in the monolithic
hospital environment. There's going to be new use
cases around. How do our electronic health
records get connected to one another so doctors can
make far better decisions. All of these things are
gonna require new enterprise software to power these
businesses and these industries.

In the media space as an
example. You know, you have a world
where an industry is going from really linear

programming. So whether that's
television, or that's music, or whether
that's movies, there's sorta very linear supply chain
oriented business model. Where a film gets made, it goes to the movie theater
for three months and then afterwards it goes to
iTunes and these other platforms. To a world where people want
experiences on demand. So that's gonna change, how distribution works on
the scale of 3 billion people that are on the
internet.

And again, no media company
has a platform. That is gonna be able to
actually power, how content and data and information moves through
the system at scale. We were just, I was just in LA yesterday
meeting with a media company that has basically
done predictive analytics. To find their potential
movie goers in the population of 3 billion
internet users. They wanna be hyper targeted
on how they get to the very, the sorta specific 30
million people that are fans of different kinds of film
types.

And so all of a sudden, you
have a movie company, that actually needs big
data. And they need business
intelligence and they need marketing
automation. To go power how they're
gonna go market and distribute their movie. So again. Totally unpredictable
mash-up of two industries coming together,
where all new kinds of software is gonna be
necessary. So, every industry is going
through some form of this change. You can pick any industry
you want, and take and sort of zoom into it and
say, what are the underlying technology factors that are
gonna change the business model of this industry in
the next couple of years?

And then there's gonna need
to be software that goes and powers those experiences. Cool, are we all on the same
page, great. I just need a drink water,
so if you could say something
that would be great. >> Can we also edit out my

water drinking, so it doesn't make it look like
I have some weird water
problem? Okay, cool. I mean, think about the
future of water. Who's gonna power that,
right? >> Gonna need, that's gonna
need software, I'm sure, so.

>> So basically, every
company in the world. We think of, you know, the
good fortune of being in Palo Alto, being at Stanford
is, is we're so hyper-focused on technology. And we think of the
technology industry as an industry, but in reality
what is happening is every company,
every industry is gonna have a technology component of
what they do. You won't be able to survive
in the future, enterprises will not be able
to survive in the future, if they don't get good at
technology.

If they don't have a
competency in leveraging data and using
these new tools. But they're gonna do that
through working with what we consider the
technology industry, as opposed to everybody
building up these pockets of expertise themselves. So there's gonna be a lot of
partnership over the next five to ten years, where
companies are gonna need to use technology to work
smarter, to work faster. They're gonna need to do
this more securely, and this is gonna change, not
only how individuals work in these environments, but
ultimately change the business models of these
companies.

So that was chapter two of
this. That's all the things that
have been changed. Now some practical advice, where to kind of help you
get started. And to be fair, most of this
advice is looking through the sort of
lens of retrospect, which means that, that this is not
always how things happen. But I can look back in time
and say, these are things that led
certain things to be true. So it's hard to be super

deterministic about building a company, and you might not
have all of these things totally figured out.

But these will give you a
sense of some of the patterns to recognize
and opportunities to exploit as
you're building, or thinking about building, an
enterprise software company. So the first is, spot
technology disruptions. And this is gonna be true whether you're building
consumer or enterprise. The rest are more enterprise
oriented, but this is just fundamental to
entrepreneurship, if you're gonna build a tech
company. So you have to look for new
enabling technologies or major trends, like
fundamental trends, that create a wide gap between
how things have been done, and how they can be done.

And so, looking back in time
at our business, the gap was basically, storage is
getting cheaper, internet was getting faster, browsers
were getting better. Yet we're still sharing
files through this very complicated, very
esoteric, very slow and cumbersome means. So any time where the delta
between what is possible and how things work today is at
its widest, that is an opportunity to go
build new technology to go solve a problem. And so as you're looking at
the enterprise, the question is, what about
the cost of computing dropping so rapidly changes
what enterprises can do with their data?

What does it change about
what you can do from a business model
standpoint? What was impossible, because
of either economic feasibility or
technical feasibility 10 or 15 years ago, that now all
of a sudden is possible. And a fun thing you can do
sometime is, if you go look at articles
from like the 1990's or even the 1980's, business
articles about technology. All we're really doing is
repeating a lot of the technologies that were
tried 10, 20, 30 years ago.

It's just, it was too
expensive, it was too unusable, and we
didn't have the enabling technologies to
make it possible. So you can kind of see this
pattern emerge constantly where something that was
impossible five or 10 years ago, all of a sudden becomes
very practical. So I'll give you an example. There's a company called
PlanGrid. Does anybody know what
PlanGrid does? Okay.
Cool. Are you in the construction
industry? You are? >> Yes, I am. >> Oh, my god.
What does that even mean?

>> In terms of PlanGrid or
like in construction? >> Construction. >> I work at a job site. We build buildings. >> Holy crap, that's great. >>
Okay, so. Thank you. So basically PlanGrid is a
mobile application that lets you manage construction
projects. Lets you get access to your
blueprints, lets you manage all of the data around a
construction process. And what this team realized
was that $4 billion, I believe, are spent every
year printing out blueprints and making all of the
prints, and updates to those blueprints
any time there's a change.

And then that has to ripple
and cascade through a very wide
network of contractors and construction people every
time that there's even just one
slight, minute change. And they realized that, you
know, with the iPad, all of a sudden, we have the
perfect form vector to be able to load up blueprints
and load up content. And this is something that could ripple through the
construction industry, which isn't necessarily
notoriously known for high technology, except for
on the design side.

How could they build
technology that would make the data access, the
data collaboration problem really, really seamless and
easy to do, in a very, you know, traditional
industry that hasn't quite changed for a

while. So it was a sort of perfect,
sort of discovery of a technology change in a
market and figuring out how do those
two things converge. And then this team built a
great startup out of it that is doing incredibly well,
and totally taking over the construction industry as
proven by this individual.

Thank you. >> So the next thing is, in
the enterprise you want to start intentionally small. What I mean by that is, you
want to find, and this is again, true of a lot
of companies, but even more true in
enterprise, in a user-led paradigm. You want to find the wedge
that is sort of the very natural place where you can
create a product that will slip in between the gaps of
other existing products. But is something that you think over time expands to
become a more important part of the
enterprise architecture.

So, what you want to start
to do is to say, we're gonna take this sliver
of a problem, we're going to totally make
the user experience on that incredible. We're gonna either change
the business model. We're gonna create new
technology that makes this previously difficult problem
really, really simple. And it might feel small at
first. Maybe you're going after
small businesses and then you're gonna move
upmarket. Maybe you're starting it
just a sliver of the use case and then you're
gonna expand out.

But you intentionally start
small because you will not be able to compete
with an incumbent. Because the incumbent is
gonna always go for the full solution. So you have to find what are
the gaps in the full solution that are
significant enough where a customer is going to want to solve the problem
with a discreet technology. But that over time you're
gonna be able to expand, again either to larger
customers or to more use cases over time.

Great example is zenpayroll. So zenpayroll, started by a Stanford graduate or Stanford graduates, couple of years ago. Basically discovered that the payroll process in small businesses is an incredibly complicated and annoying and cumbersome process. And that's because we use the same vendors that we've been using for decades to do that. And they weren't sort of digitally oriented. You didn't get your payments as a receipt, over email. It was a very complicated. You didn't really get to see graphs of your salaries.

There was no real good data around this. And they said, we're just gonna take off the slice that is most painful to startups around hiring people and around paying people, which is just that payroll management process. We're going to plug into a lot of existing infrastructure, but we're going to make it dead simple to go do this. And now they're able to move both up market over time as well as deliver new services over time. And what happens is the incumbents in this market initially look at something like this and payroll and they say, well, that's small.

It's only for small businesses. It's not gonna be very powerful. But that's just the start. Because as they get that wedge and as they fit into the market, they're gonna be able to expand again over time, build out more services and more capabilities. But they found just the exact right opening to build a new company and have that emerge. The next is you really want to find asymmetries. You want to find, my God, okay, so can we do like some auto like some tuning of a voice later so at least it'll sound like I'm doing rap.

{do things that incumbents can't or won't do because it's economically or technically infeasible} Okay, so you want to be able to, you want to do things that incumbents

can't or won't do. Because, either the economics don't make sense for them. Or the economics are so different, or so unusual or because technically they can't. I'll give you two examples. So, if you're gonna build software today for the enterprise that goes after an incumbent category that has more of a sweet oriented approach.

Then what you wanna go do, is you wanna build technology that's gonna be platform agnostic. Because what sweet players will do, is they want everything to be integrated with itself and that there's more value with that vertical integration. But you wanna go after a different access, which is, you want your technology to work across all of the platforms. So that way, you can work with so many different kinds of customers, and you can be an ally to so many different kinds of platforms, which a traditional incumbent is not able to do.

So that's something that is usually technically infeasible, because of how either something has been architected. Or because the fundamental component of the business model to not do that as the incumbent. But, the other thing is try and do things that are economically infeasible. You can look at the cost structure of an incumbent company and discover where are they not going to be able to drop their prices. Because that business model is fundamental to the existence of the company.

Or where can you find ways of monetizing the customer that are unusual or unique, that no one has discovered before. And thus is really not practical for anyone else to do. That's a company called Zenefits, where they have an HR management software company that helps you, as a small startup, manage all of your benefits, all of your HR information. And instead of charging the startup, which might not value that software at the

stage that they're at.

They realized that they
could get commission from the
insurance companies. And that basically pays for the ability to use their
software. So, the customer itself is
not paying for Zenefits, the Zenefits
platform is being paid for by the insurance company. And they've thus invented a
business model, that no other software
company has been able to think of or attack. And they're equally going
and disrupting a category that
hasn't seen a lot of innovation previously, which
is the health and benefit space within small
businesses.

The next is you wanna find
the really crazy but, but sort of still somewhat
reasonable outliers within the customer ecosystem. So you need to find the
customers that are sort of at the edge of their
business, or at their business model, or
of their industry. And find the unique
characteristics about those customers, and leverage them
as your early adopters. So Paul Graham has a great
article, where he talks about living in the
future and building for what is missing when you're
living in the future.

So that's a sort of easy way
to spot trends and patterns about disruption
that's playing out. The same is true in the work
place. If you find customers that
are working in the future, you will be able to work
with them to find what is missing when you're working
in the future. And how we build technology
that supports all these new use cases that are going to
emerge. There's a company called
Skycatch, which does enterprise
drones.

And at first it seems kind
of weird, but in the construction space,

in farming, in oil and gas, they're using drones
now for data capture, and the ability to do modeling
of, of different environments. And so, this company is able
to find all of the companies that are at the bleeding
edge of their industry, what is unique or new about how
those organizations operate. And they built and worked
with a lot of those early adopters to establish their
platform, which is again, really one of the first
enterprise drone companies.

So the idea is go look at
your market. Find the customers that are
at the bleeding edge of the market, who use
technology to get ahead, and that use technology for
performance advantages. And then go work with them
to discover how your product can evolve. Two minutes. Okay. Good.
Right, so. The next one is.
And then there's a part two in
five minutes from now, or is that not the case? Okay, so, listen to your
customers, but don't always build exactly
what they're telling you.

This is a really key distinction around building
enterprise software. Your customers are gonna
have a large number of requests. Your job is to distill those
requests down into the ultimate product. That does not mean that
you're going to build exactly what they tell
you to build. It is your job to listen to
their problems, but actually translate those
into what is going to build the best and simplest
solution to those problems. It's really your job, and
Palantir does this, as an example, with very,
very complex issues.

And then they distill those
down into simple solutions for very, very, complex problems that the
customer otherwise would not have known how to ask for You want to modularize, not
customize. So build a platform, as
opposed to building all of the custom technology and
custom vertical experiences directly in the software
itself. Make sure that you really
think about openness in APIs as a way of delivering

vertical or custom experiences. Don't build that directly into the product.

Focus on the user always. So, again, the magical thing about building an enterprise software company right now. Is you can keep consumer DNA at the center of the product. That will always mean that adoption is easier. Your product has a much higher chance of going viral. It becomes easier to sell into the organization. So always make sure you bring consumer DNA into the product development process. And your product should sell itself, but that does not mean you don't need sales people.

So this is a really important distinction. Leverage everything about the internet, leverage everything about users to get to your customers. But you still will likely need sales as a way to help your customers navigate your product. Help your customers navigate the competitive landscape and the ecosystem. So you're going to want consultative, very, very domain-oriented sales individuals, that are gonna be able to be helpful in your customer being deployed and enabled in these organizations. But don't make that be a substitute.

Do not make that be a handicap for not building a great product. So you fundamentally have to keep the product at the center of that. Company called Mixed Panel comes in through the developer, and then over time is able to sell to that organization with a more inside sales-led process. Also read these three books. So Crossing the Chasm, Innovators Dilemma, and Behind the Cloud. These three combined, especially if you read, if you binge and you just read them all.

You'll definitely come out the other end with a massive software company. And so, in closing today. Right now,when you leave this room, is an amazing time to go start an

enterprise software company. So I wish you the best of luck. And of course, if it doesn't work, we are hiring. So, and then the only other caveat is please do not compete with me. Because I have a lot of competition already. So ideally either come work for us, or build your own company.

So thank you very much.