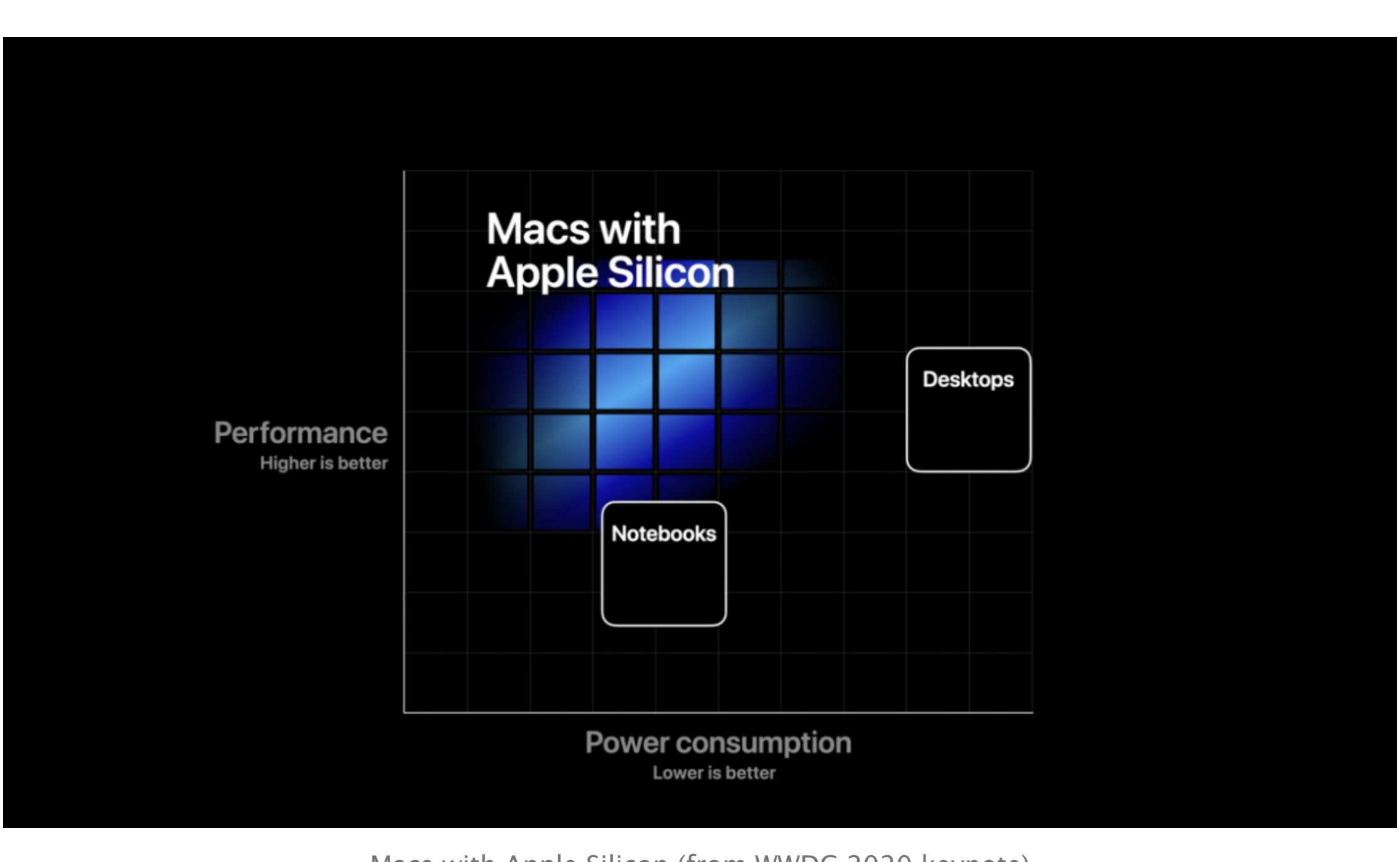
Apple's Relentless Strategy, Execution, and Point of View

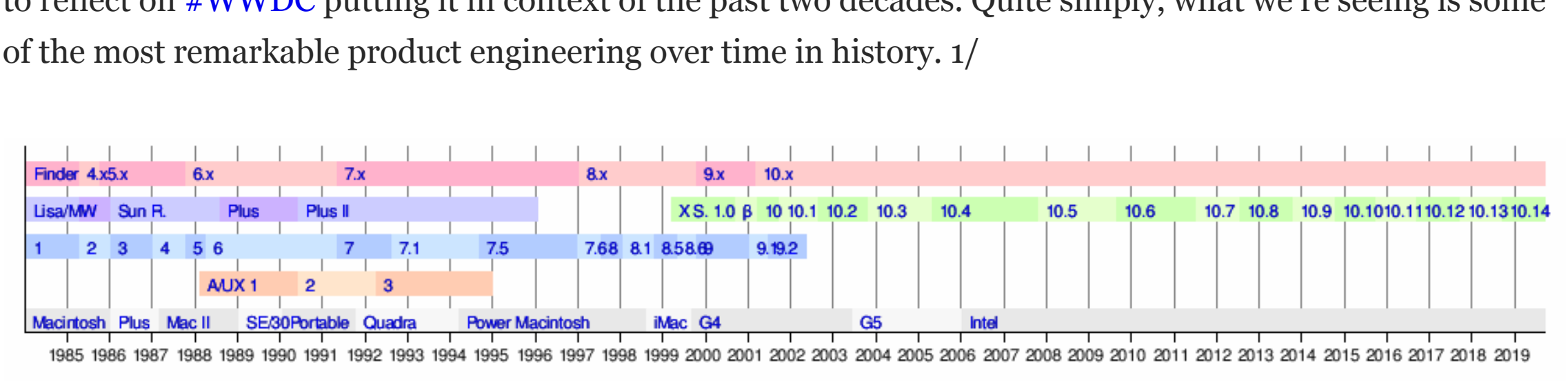
Steven Sinofsky
Jun 27 · 14 min read★

Apple's announcement of "Apple Silicon" is important for many reasons. Delivering on such an undertaking is the result of remarkable product engineering. An annotated thread…



Macs with Apple Silicon (from WWDC 2020 keynote)

Amidst all the details, installing pre-release, and commentary (including my own) I want to take a moment to reflect on #WWDC putting it in context of the past two decades. Quite simply, what we're seeing is some of the most remarkable product engineering over time in history. 1/

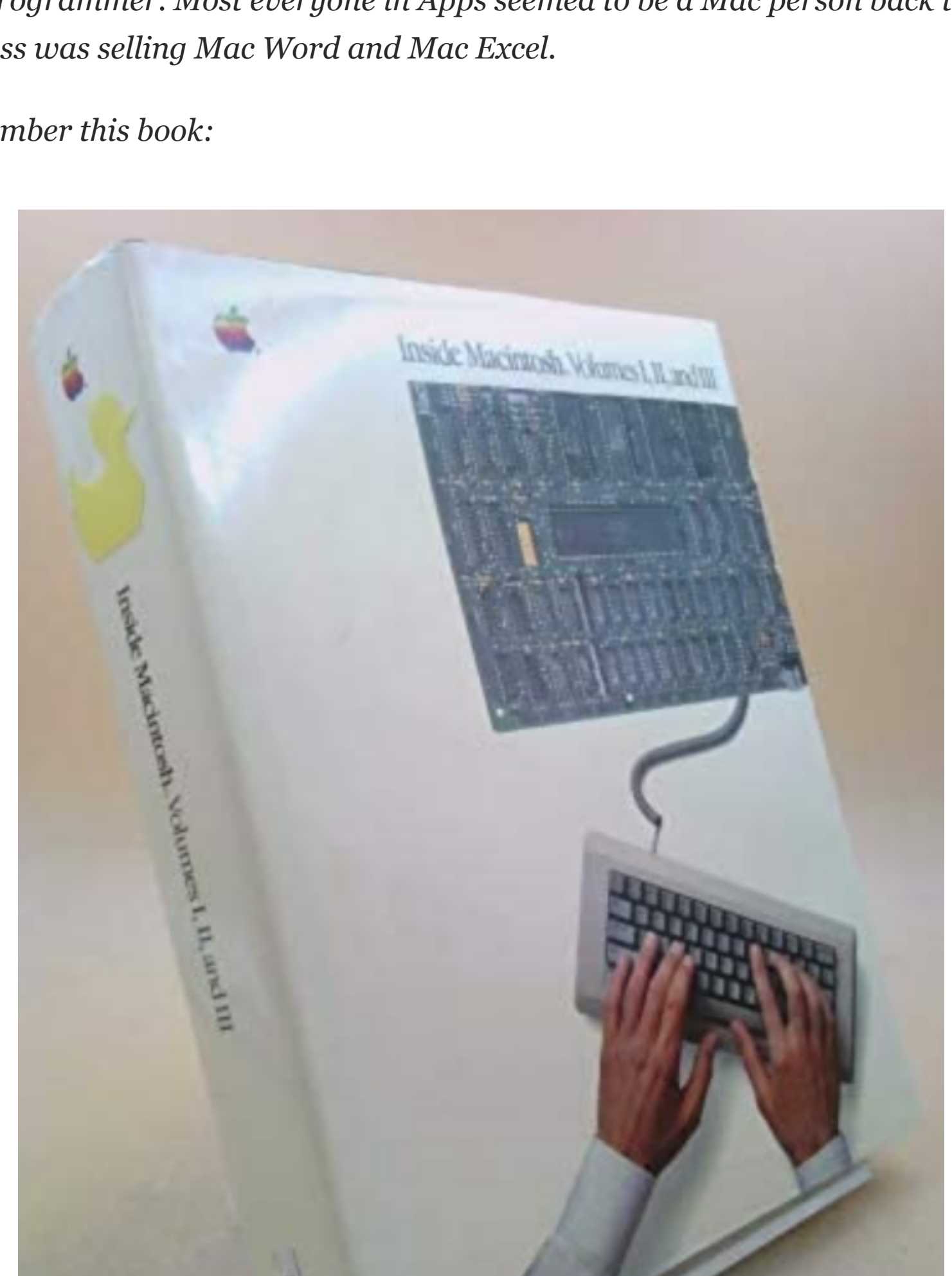


2/ It is easy to get wrapped up in debates about specifics, excited by tweaks or surprises, even an occasional scandal, or to wonder about the quality (is this is a good beta?). Under the hood, is a team that over time has done more and executed better than any I can name, ever.

3/ Having walked in similar shoes for many years, and importantly starting from when Mac was a hammer smashing through a screen, through the lowest lows (fine, we'll keep doing Office), resurrection, and reinvention, it's wild for me to consider what makes it so amazing to me.

My first Mac software was an app called "Mac Mendeleev" which was a visualization tool for over 50 periodic properties of elements. It was written in Pascal before the Mac Programmers Workbench of MacApp existed. Like so many people hired into "Applications" at Microsoft, I was hired from college as and to be a Mac programmer. Most everyone in Apps seemed to be a Mac person back then. Half of Microsoft's business was selling Mac Word and Mac Excel.

Many might remember this book:



Inside Macintosh Volume I, II, III from Addison Wesley

4/ Many have their own ideas for why Apple is doing so well, but I want to share what I think of as what got them to today from a kind of unique vantage point. I'll focus on making the products, not to take away from marketing, manufacturing, selling which all have equal roles.

5/ Three factors continue to blow me away (for lack of a better expression), especially after today.

- Fearless multi-year strategy
- Clear unified planning/prioritization
- Wildly unprecedented execution

Many tend to focus on "strategy" alone (such as vertical integration), or "manufacturing execution" (eg Tim Cook's legendary supply chain work when he joined the company), or "marketing" ("Get a Mac"). All of those on their own were amazing and critical. I am choosing to look at the overall arc of product development as that underpins everything that I believe is singularly unique.

6/ My jaw dropped when Tim Cook discussed the transition to Apple Silicon (ASi) as a two year journey. First, that's like no time at all. Second, that's an incredibly long time to tell everyone how long it will take and that they should be patient. Seriously.

To put this in context. The transition from PowerPC to Intel was announced in Spring 2005. The first PCs shipped in January 2006 and by August the whole line switched to Intel (about one year). By Summer 2009, OS X no longer supported PPC and by 2011 Rosetta emulation was entirely gone. The entire transition looks like it took 5 years on paper, but in reality the biggest thing holding up was Adobe. I believe had Adobe been on board sooner the transition would have been accelerated (rather than delayed as it was). Much of that feeds into the timeline we saw discussed this week, including Tim's "2 year" time scale.

7/ But really that is incredibly brave when so much could potentially change, more importantly could go wrong. Every big company does multi-year planning (I did) but everyone knows those plans mean little after a fiscal year. Apple is entirely different in that regard.

8/ The big thing about this is how Apple's overall model of enables this to work. Every aspect of the system has to come together to create an environment where choices can be made AND supported that allow these plans to have integrity.

What I mean by Apple's model is not about its direct to consumer business or vertical integration, but the culture of having a "point of view." Apple makes products that customers love and are delighted by, but it makes them by studying technology, the market, and usage to arrive at plans and strategies. Unlike what you read in textbooks, Apple is much less about responding to micro changes, hype cycles, or even "feedback." In fact you can see often how Apple's model does not work so well when it rushes products to market or listens too closely to hype (eg Home Pod). Apple is a company that has a point of view — when the point of view lines up with a great product people love, it can become an unstoppable force.

9/ For example, while Office reliably shipped for decades planning with Windows was super difficult because Windows had a different view of planning and shipping. Plus enterprise versus OEM customers. It was a miracle when we got the summer of '95 done.

Here is where the difference in distribution channels totally changes how products are made. First, in the 90's Office optimized for releasing products to retail channels, which meant a large number of country specific "customers" each requiring lead time, including time to develop print advertising, and more. Windows focused on OEMs which work with even longer lead time and also had seasonality (back to school, holiday, spring refresh) that required working backwards as much as 3 or 4 months. Lining all this up became futile when in fact the Windows project completion date was a moving target. This is the downside of teams with business goals pulling in different directions and working with an ecosystem of large independent companies as partners. The upside, however, should be obvious.

10/ The transition to NT started in 95, was supposed to take a year. Windows XP was 2001(!) So we ended up shipping ever decreasing quality products we (eg Office) had to support for that time (9x). Still Win2000 was supposed to be "it", then XP, then XP SP2, delaying all else.

Note: There was an org chart divide between Windows NT and 95 that needed to be resolved as NT was developed as the server strategy and in an isolated org (based on a smart choice in hindsight, but one that only postponed the reconciliation of strategy). The idea was to slide the NT kernel under the Windows 9x kernel, but as it would turn out that was not so straight forward. As a result, the 9x team shipped 98, 98 SE, and Me to continue to evolve the hardware ecosystem and try to keep the momentum going. Windows 2000 was the first reconciled kernel+GUI except it fell short in compatibility. Finally Windows XP in 2001, but that had significant security issues at launch. In the interim Office shipped, Office 97, Office 2000, and Office XP, and Office 2003 — all of which were 32 bit apps though it wasn't until Office 2003 (shipped August 2003, a year before XP SP2!) that we finally stopped supporting the 9x platform (8 years after it released).

11/ Example: the 64-bit transition. It took 20 *[ed., 25]* years for it to happen. 20 *[ed., 25]* years. It still isn't done. OTOH Excel 2.2 still runs on 32 bit Windows, so does WordPerfect MS-DOS, which is a miraculous achievement AND honestly highly valued by HUGE customers. Different worlds.

It is rather incredible how different the 64 bit transitions were for each company. Microsoft pioneered 64-bit computing — by working with AMD it drove the compatible industry standard that in a sense Intel was fighting (to avoid commoditization of their IA64). Yet the "patience" shown in maintaining compatibility has been remarkable. Even today, Office installs the 32 bit product by default and recommends it. Our team made that choice in 2003. It is still the case [Correction: That changed according to this post https://techcommunity.microsoft.com/t5/deployment/office-proplus-and-office-2019-now-install-64-bit-as-default/m-p/363394 *in spring 2019, though my own experience and Surface Pro that I bought last Fall disagree.]*

Apple began requiring 64 bit apps in 2017. Two years later 32 bit apps were no longer supported. [Error. I should have also said that the equivalent of Apple starting 64-bit was in Snow Leopard in 2009, meaning the entire development was a decade. I incorrectly focused on the transition time. Microsoft only just announced a transition for application ISVs to 64-bit only.]

An important part of the Apple model is "fearless" in that Apple is willing to turn over its partners and ecosystem members to new ones in an effort to stay on strategy. This is a key benefit of having committed consumers as everyone's customers — consumers replace devices on their own. The enterprise computing model does not let you turn over customers, because there are only 2000 global companies.

NT to go 64-bit with Intel

By Bob Francis

Microsoft Corp. plans to take Windows NT on a 64-bit road trip to Intel Corp.'s Merced processor, which is expected to come out in 1998, the two companies said recently.

The announcement means information systems managers can expect to see some 64-bit NT applications when the new chip is available. When the 32-bit Pentium Pro chip was released, Windows NT was available, but very few 32-bit applications were ready for the chip.

"Intel has been very successful in developing good low-cost processors, so we're certainly interested in seeing how this will work," said Michael G. Tardif, vice president of enterprise technology at Goldman, Sachs & Co.

Value In NetWare

Other IS managers weren't so sure they would need systems that powerful. "We still see a lot of value in the NetWare environment," said Diane Delvecchio, an IS consultant at Smith Environmental Services, Inc. in Dallas.

Potential uses for the 64-bit chip and operating system include large databases and high-speed graphics applications.

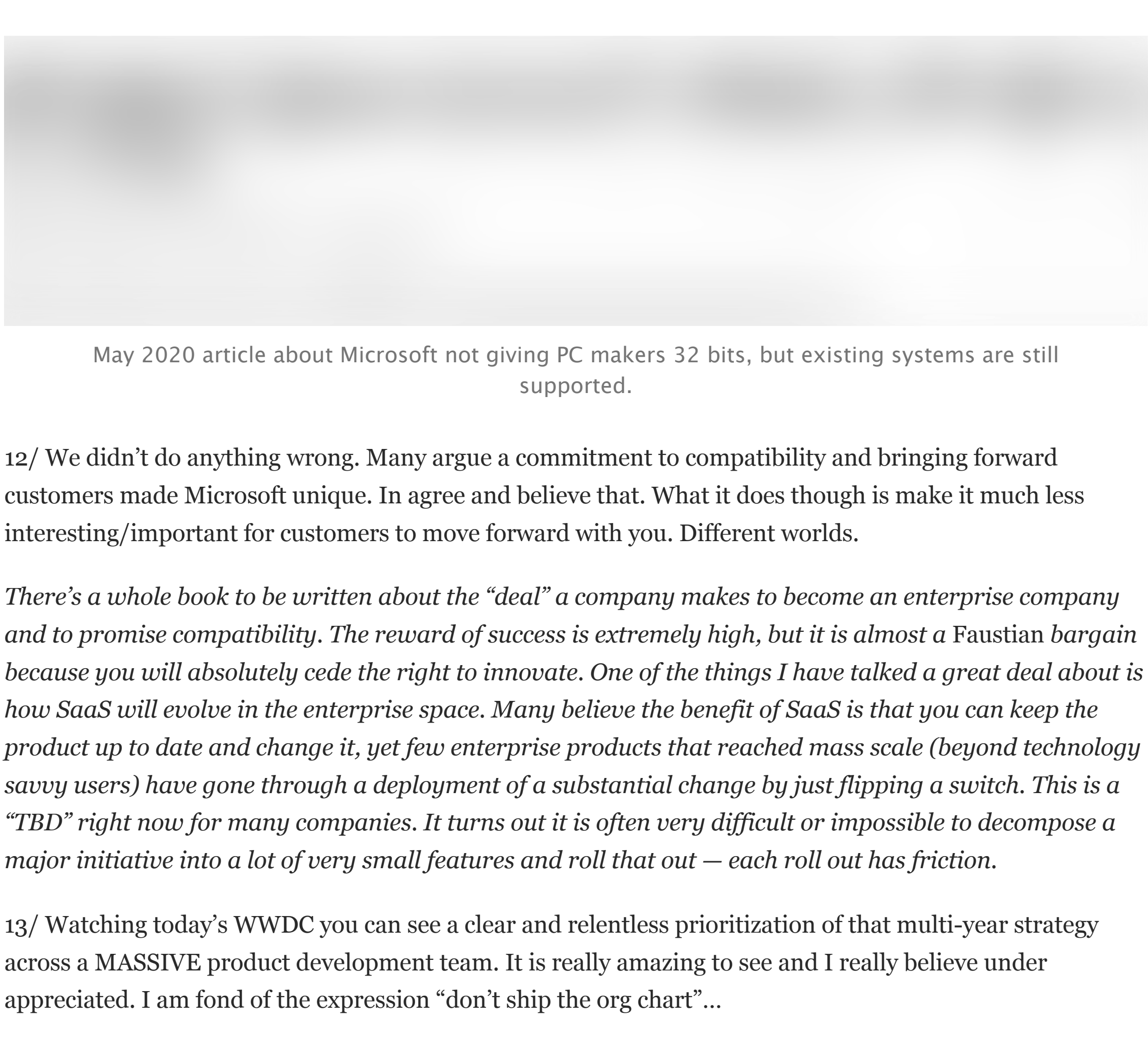
The Merced processor is Intel's next-generation 64-bit processor, which is part of the company's next generation of chips. Processors from Intel in Santa Clara, Calif., are currently 32-bit; they process instructions 32 bits at a time. The 64-bit chips will double that speed. Merced is expected to be available late next year, with general availability in 1998, sources said.

Intel is on track with the Merced processor, which it is co-developing with Hewlett-Packard Co., said Dean McCarron, an analyst at Mercury Research in Scottsdale, Ariz. "They've been very pleased with what they'd done so far, but there can always be slip-ups with new chips," he said.

But Microsoft is plunging ahead with the 64-bit version of Windows NT. The company will release a preliminary copy of the 64-bit Windows NT specification at an upcoming Professional Developers Conference, company officials said.

COMPUTERWORLD SEPTEMBER 30, 1996 (www.computerworld.com)

ComputerWorld from 1996 announcing intent to work on 64 bit Windows.



May 2020 article about Microsoft not giving PC makers 32 bits, but existing systems are still supported.

12/ We didn't do anything wrong. Many argue a commitment to compatibility and bringing forward customers made Microsoft unique. In agree and believe that. What it does though is make it much less interesting/important for customers to move forward with you. Different worlds.

There's a whole book to be written about the "deal" a company makes to become an enterprise company and to promise compatibility. The reward of success is extremely high, but it is almost a Faustian bargain *because you will absolutely cede the right to innovate. One of the things I have talked a great deal about is how SaaS will evolve in the enterprise space. Many believe the benefit of SaaS is that you can keep the product up to date and change it, yet few enterprise products that reached mass scale (beyond technology savvy users) have gone through a deployment of a substantial change by just flipping a switch. This is a "TBD" right now for many companies. It turns out it is often very difficult or impossible to decompose a major initiative into a lot of very small features and roll that out — each roll out has friction.*

13/ Watching today's WWDC you can see a clear and relentless prioritization of that multi-year strategy across a MASSIVE product development team. It is really amazing to see and I really believe under appreciated. I am fond of the expression "don't ship the org chart"…

14/ Shipping an org chart is one of the greatest forces one must work against in any team of more than 100. We're talking a product "team" of (maybe?) 20,000. That's twice as big as anything I did and more like the size of MS when I left.

I have no idea how big Apple's R&D team is or how many work on what we saw at WWDC. For many years all of Apple was the size or smaller than the Office or Windows teams I was part of. I watched this closely as a competitor. Now Apple has such breadth, like Microsoft, it is difficult to compare without inside details.

15/ It is incredibly clear that everyone at Apple puts strategy requirements above anything "local". When you wonder why there isn't more new in Notes or why Mail is missing stuff it's because supporting a multi-year strategy trumps individual teams and that's a good thing.

16/ To execute requires everything in the company to operate as though strategy matters most. It means communication. Performance reviews and rewards. It means management top down reinforces it and isn't "random" or "inconsistent". That is so difficult internally. Painful even.

17/ My experience was so different. Again not worse but different. Microsoft operated much more locally and hence was far more resilient, in many other businesses, and served many different customers types. Some would even say more responsive to customers. Activity v. progress?

18/ In BigCo when you're asked to do things that don't seem "as" relevant to your success, it is a tax — when one group "requires" another to do something. Ppl at a BigCo (not Apple) know what a tax is. Apple operates like taxes are good. That's unique and worth appreciating.

A way I think of this is that, as an example, Windows was broken down into 50 or so teams. Each team owned an area of the product (Office ran the same way about 35 teams). Each team was about 50 engineers. You could think of a team as media platform or device support. At some point you balance resource allocation with continuity for individuals AND also maintaining expertise. This means at some point you will have a tough time figuring out what every member of a team might do if you don't need or want something from a technology area in a release (eg maybe stability is good). One way to do this is to minimally staff a team to only offer what is required at the strategy level, which is one way to balance those conflicting goals — which is why the team does not act like a "product team" and have their own research and goals, they simply work on the strategic goals. In this era of WFH and remote, this challenge is going to be front and center much more than I believe people are considering.

19/ There was so little "random" stuff today or most releases. That really matters — it means in 5 years there's nothing to clean up, nothing to obsolete causing pain with no value. The hardest thing for me was getting rid of something no one used (MediaCenter!)

It is literally impossible to end something at Microsoft. I can't tell you how many times I (or we) tried. A running joke was how long the Excel team kept doing bug fixes on OS/2 Excel for a European bank. It was long after no one used OS/2 (except that bank). Media Center was a classic example and when I tweeted that I knew I would get tweets from people saying "I used it." We even published the usage data which of course no believed :-)

20/ Finally, the image in (1) is the Wikipedia roadmap of Apple releases since 1984. Every single person reading this should look at it and marvel at a work of art. No company in software has done so much, so regularly, for so long, and certainly not at billions.

21/ This isn't scrum or agile or… — most would call it waterfall BUT IT IS NOT. It is planning, iterating, prioritizing, discarding, restarting, and more. I argued most of my career that having a strategy and prioritizing is the only way to execute to have this impact. QED.

Many pushed back on this. Very quickly every dialog about methods turns into a discussion of definitions or "not doing it right." What Apple does is unique — it is a long term goal, broken up into short term AND useful steps that go to market, and a relentless focus on not going off plan. They can do this because their view of the market is the same as their point of view — there's no fear of being confused by hype cycles, movements of competitors, etc.

Microsoft has a long history of making products over three releases "finally" work. That is quite different. The two biggest strategic bets Microsoft made, in my view, were on the graphical interface which started around the same time Apple began (with Multiplan being the first product, and Excel the first major new product). You know a bet is strategic if people quit over disagreeing and that happened! The second bet was on building an entire new OS for the future. What's so interesting about the bet was how the convergence of that bet with the client GUI bet (Windows 9x) was that it came almost "later". This was decidedly not Apple-like and much more about responding to the changes in the product. NT was a product that had the same relentless focus on execution as Apple did. Aligning with the client impacted that ability.

22/ One look at any other company that tried to do pre-committed release cadence maybe a company does well, but slows down. Or the scope dramatically shrinks. Or quality declines. Apple had ups and downs but look at that! I know big ex-apple execs that count this as *the* thing.

I spent a lot of time marveling at the execution of Apple and after I left Microsoft I had a chance to talk to some ex-NeXT and ex-Apple people and it was awesome to hear how much they did to make that happen. Amazing.

Some ask why Microsoft did not commit to yearly releases. First, the history of Windows was, unfortunately, not even being able to get one release done on time (except Windows XP which did make the August 2001 release, but that was hampered by the need to retreat to doing a security service pack that went from a 3 month project to a 30 month.)

Beyond our own capability to execute, a real challenge was the ability for the ecosystem to absorb regular releases. This included the end-users who had bought a PC year earlier and not having a "free upgrade" model (so it was economics). Apple famously made the OS free which was quite a conundrum at the time for us.

In Office we released on sort of 24–36 month schedules because the enterprise business did not want that much new software. Even though customers were on ARR payment (known as enterprise agreements) they did not have the bandwidth to deploy and train their employees.

Taken together it meant that at any given time only about 1/3rd of customers were on the most current software. The other 2/3rds were on software that might be 6–8 years old. That's why compatibility becomes important. It means if a team was developing something today it needed to work on an 8 year old computer.

Or did it. As it would turn out we finally had a breakthrough in this in coming to grips with the fact that an 8 year old computer was also not a computer that would ever get new software. So why bother? Well, with an enterprise agreement customer all they have to do is ask. Microsoft was in fact beholden to these customers.

23/ Before everyone cargo cults these concepts the thing about them is that they work and are magical because of two things. First, the people at Apple are amazing. Yes everyone says that and believes that but boy everyone at Apple is the world's best at their thing.

I mean it.

24/ Second, many think vertical integration or design focused or some attribute is it. Too many proof points exist as to why those aren't enough. Sony focused on design. DEC was vertically integrated (they made floppies). Even IBM. Vertical is a means to an end, not a cause.

There's only one Apple. Don't try to copy. Read the stories just for inspiration though. A great recent book is Creative Selection: Inside Apple's Design Process During the Golden Age of Steve Jobs *by Ken Kocienda.*

25/ Anyway, I wanted to end today sharing why what Apple is doing not only makes great products for a billion people but is doing so in a way that sets the highest bar for *how* things are made and what it looks like for me. // END. Discuss.

Wow there is so much more. But this is too long an annotated thread. Someone should write a book.

Finally, this post discussing historical engineering process has nothing to do with the stock market, investing, or anything recommendations along those lines.