Industry Surveys

Software

JANUARY 2023

John Freeman
Equity Analyst

CONTENTS

Contacts

Sales Inquires & Client Support
800.220.0502
cservices@cfraresearch.com

Media Inquiries
press@cfraresearch.com

CFRA
977 Seminole Trail, PMB 230
Charlottesville, VA 22901

Contributors

Raymond Jarvis
Senior Editor

Atifi Kuddus, Geraldine Tan
Associate Editors

CHARTS & FIGURES

NEW THEMES

What's Changed: We continue to have a positive outlook on the software industry for 2023. Read page 9 for our recap of 2022 and page 10 for our outlook for 2023.

What's Changed: What were the top M&A deals for the software industry in 2022? Check out pages 11 and 12 to find out.

EXECUTIVE SUMMARY

Our fundamental outlook on the software industry remains positive. While inflation, higher interest rates, a lower appetite for risk, and a strong dollar have all had a negative impact on top-line results and growth of many software companies, the fundamental value proposition of enterprise software has only strengthened. This is clearly true for cloud-based providers of applications used to manage the means of production, such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM), and for developers of the most compute-intensive applications deployed on-premises, such as those involved with design and simulation of a large and rapidly growing number of physical products, including drones, batteries, ergonomic chairs, hydroelectric turbines, cancer drugs, affordable housing, and microprocessors.

No doubt, the severity and duration of the present downturn caught us by surprise, but it has done nothing to alter what we see happening, not just among enterprise software companies but in business and the economy overall: increasingly powerful software continues to substitute for labor and dumb (i.e., not software infused) capital, freeing up human cognition for greater productivity, creativity, and generation of economic value. The pace of that substitution and the velocity of software-based innovation -- and, these days, it is difficult to find innovation not based on software -- continue to accelerate globally. We have yet to see anything that has altered the trajectory of this overall trend. With TSMC announcing at the end of 2022 that it has started volume production for its 3nm process, yielding processors with over 100 billion transistors, it is clear that the crucial fuel for software innovation and proliferation, Moore's Law, is far from over.

However, the economic environment in which this process is taking place is quite different from how it was last year. Then again, a year before that, we saw one of the most unprecedented economies in modern history in 2020 due to Covid-19, a year from which the global economy continues to experience reverberations. To say the least, the global economic situation was not helped by the unnecessary and draconian lockdowns mandated simultaneously without an effective large-scale vaccination effort by Xi Jinping and the Chinese Communist Party (CCP) in China. On top of that, Russia's invasion of Ukraine created further economic disruption and put the global geopolitical order at much greater risk.

In this environment, enterprises are, as expected, looking to cut costs, which include overall software spending. Yet, unlike other areas of tech where revenues can swing more wildly, we still expect positive Y/Y growth in 2023 as cloud migration accelerates, resulting in legacy maintenance revenue falling more precipitously, as has been reflected in recent results from the likes of Oracle, Open Text, and Citrix. In fact, we expect the pace of cloud migration to accelerate as enterprises, while facing inflation and recession in quick succession, pursue greater operational agility and seek out more permanent cost-cutting solutions and potential competitive advantages found in software.

According to projections made by tech-focused market research firm IDC**, total global software revenue grew 6.5% Y/Y in 1H22, down from 10.7% for full-year 2021.** IDC expects similar growth in 2H22 when those numbers are reported over the next few months. We now expect a Y/Y growth rate of 6.3% in 2022, a tad lower than IDC's. We differ from IDC primarily in that we expect higher cloud growth and a greater contraction in traditional software revenue, for 2022 and through 2026.

For full-year 2022, IDC projects 15.6% Y/Y cloud revenue growth, which would be down from 20.7% in 2021 and below our 18.4% forecast. Meanwhile, **IDC expects traditional software revenue, which is mostly maintenance now, to contract 1.1% Y/Y, vs. our expectation of a 3.1% decline.** Wherever the numbers end up for 2022, both CFRA and IDC project 6% to 8% annual overall software growth for the next four years. We also agree with IDC that 2023 will be the first year that cloud-based revenue exceeds license and maintenance revenue. However, our forecasted 4-year cloud revenue CAGR is a bit higher than IDC's – 18% vs. 15% – and we see traditional software revenue declining in the mid- to high-single digits, whereas IDC is projecting traditional software revenue to be flat over this period.

CFRA **Industry Snapshot** www.cfraresearch.com

Software

Outlook: Positive

KEY TAKEAWAYS:
CFRA forecasts a considerable topline growth slowdown in 2022 for the 17 constituents of the S&P 500 Software Industry, following the surge of growth in 2H20 and 2021 that resulted from the pandemic and the accelerated migration to cloud computing and digital transformation projects among enterprise organizations to cut costs, support remote work, and attain greater operational agility. We now see significantly slower growth for 2H22 and 1H23 due to higher interest rates that discourage borrowing and the prospects of a recession.

2023 PREVIEW	
CFRA's Y/Y Revenue Growth Forecasts for the Total Global Software Industry:	6.5% for 2022, vs. our previous forecast of 7.6%; 6.0% for 2023 vs. 7.3% previously; 7.5% for 2024 vs. 8.2% previously – driven by 16%-17%+ Y/Y cloud growth, partially offset by declining traditional software sales.
Consensus Y/Y Revenue Growth Forecasts for the 17 Constituents of the S&P 500 Software Industry:	16% in 2022, which would be down from 20% in 2021 but up from 13% in both '20 and '19.
Consensus EBITDA Margin:	40.0% in 2022 vs. 44.8% in our July survey, and up from 41% in 2021, 40% in 2020, and 37% in 2019.
Industry-Focused Exchange Traded Funds (ETFs):	Invesco Dynamic Software (PSJ) iShares Expanded Tech-Software (IGV) SPDR S&P Software & Services (XSW)

RETURNS OF SOFTWARE AND IT EQUITY INDEXES	
S&P 500 Software Industry Index 2022 Return:	-25.1%
S&P 500 Software Industry Index 2021 Return:	+43.7%
S&P 500 Information Technology (Sector) 2022 Return:	-26.2%
S&P 500 2021 Return:	-19.3%

NEAR-TERM THEMES

Cloud Migration

Cloud Migration Continues to Accelerate

The pace of the migration of enterprise applications from on-premises deployment to cloud services continued through 2022, although, unlike previous years, customers have become more price-conscious, creating more direct price-based competition, as well as greater "share-of-wallet" competition among large providers. Older cloud-first providers like Salesforce.com and ServiceNow continue to take share of that "wallet", despite the slowing rate of growth. Firms where 50%+ of revenue follows a legacy license and support model – e.g., Oracle, Constellation Software, OpenText, Citrix Systems – continue to struggle to achieve positive organic growth, although we see more rapid contraction of legacy revenue accelerating in 2023 as enterprise customers scrutinize their IT spending more in 2023.

M&A

The Conditions for a More Active M&A Are Quickly Improving

Logic would dictate greater M&A activity in enterprise software in 2023 than in previous years, although we would have expected the number of deals to have ramped already given the attractive valuations and healthy balance sheets of most potential buyers. Instead, we have seen a small number of large deals such as Adobe's $19 billion pending offer for Figma, a cloud-based upstart in design and rendering, which is still pending approval. Indeed, the FTC's more interventionist stance on acquisitions and anti-trust enforcement in general is likely contributing to a more muted M&A landscape. However, acquirers such as Salesforce and Oracle are still digesting recent deals, and Microsoft and Adobe are still struggling to get regulatory approval for pending deals. We see this logjam loosening a bit in 2023 and continue to expect a larger number of smaller deals to get done.

TOP 5 BY MARKET CAP

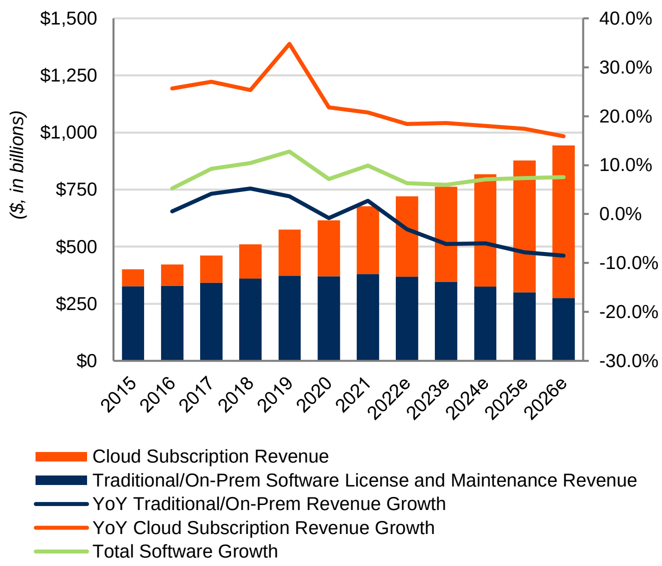
(as of Dec. 31, 2022)

RANK NO.	COMPANY NAME	MARKET CAP ($B)
1	Microsoft	1,657
2	Oracle	227
3	Adobe	152
4	salesforce	135
5	Intuit	105

Source: CFRA, S&P Global Market Intelligence.

FINANCIAL METRICS

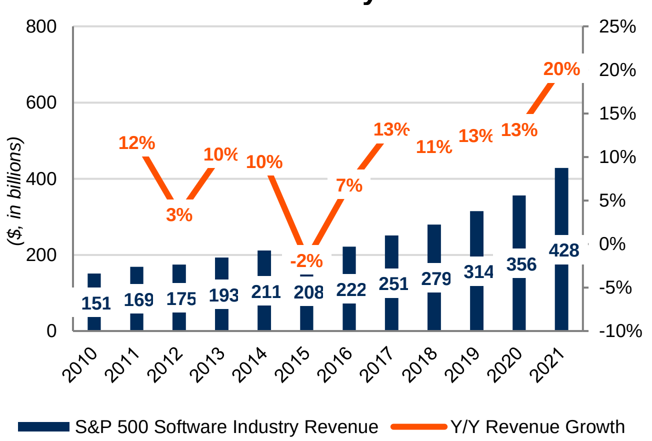
Total Software Industry Revenue*



e-estimate
Source: CFRA, IDC.

- In 2021, total Y/Y software revenue growth reaccelerated to ~10% after falling to 7% in 2020 from 13% in 2019 due to Covid-19. License and maintenance revenue from traditional, "on-premise" (on-prem) software managed to grow 3% in 2021 after falling 1% in 2020, the first absolute decline since 2008.
- We expect cloud revenue growth to fall below 20% in 2022 and remain in the high teens over the next four years. CFRA projects lower overall software revenue growth in the 6% to 8% range annually through 2026, driven by cloud and the rise of meta-software.

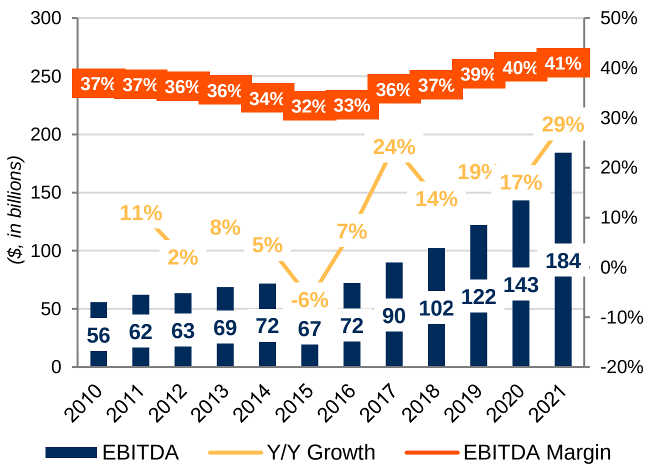
S&P 500 Software Industry Revenue



Source: CFRA, S&P Global Market Intelligence.

- Revenue growth for the S&P 500 Software Industry has clearly been accelerating since 2018, increasingly driven by the cloud shift tailwind for Microsoft and Adobe, as well as the increasing impact of faster-growing pure cloud providers like Salesforce, ServiceNow, and Workday.
- 2021 finished very strong, jumping to 20% Y/Y from 13% in Covid-impacted 2020. We will update revenue for 2022 after all full-year results are reported, but we expect significant deceleration into the mid-teens.

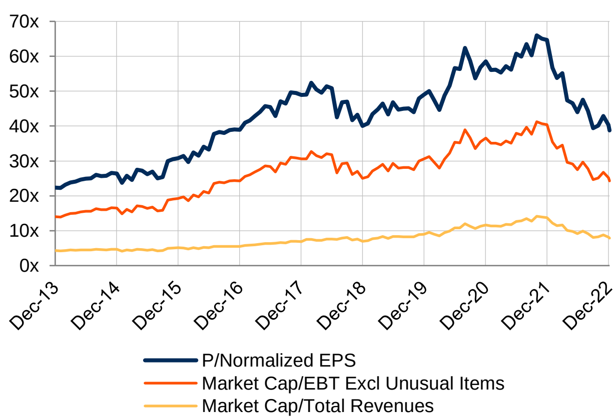
S&P 500 Software Industry EBITDA



Source: CFRA, S&P Global Market Intelligence.

- Total EBITDA margin for the S&P 500 Software industry constituents has been expanding steadily since 2016, due primarily to the growing cloud/subscription tailwind many legacy software vendors are beginning to enjoy as they shift from license sales to cloud subscriptions, partially offset by the continued high level of investment targeted at top-line growth rather than maximizing profit.
- We expect this cloud-transition tailwind on profitability to continue strengthening into 2023 despite the present economic slowdown. In fact, we would not be surprised to see EBITDA margins hold steady in the second half of 2022 as cloud-based providers pull the lever back a bit on R&D and S&M, tilting toward expanded profitability vs. growth.

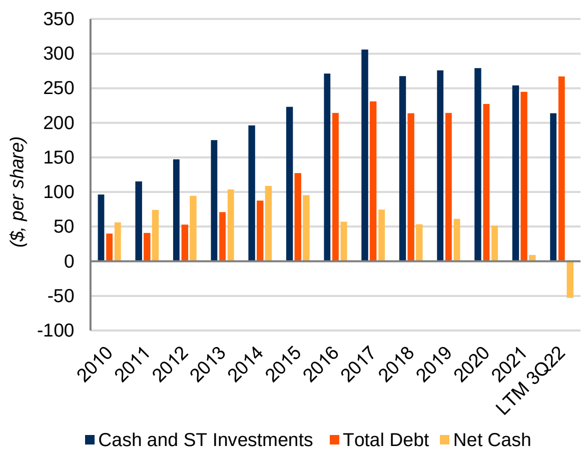
Normalized & Forward P/E Ratios



Source: CFRA, S&P Global Market Intelligence.

- Despite the recent retrenchment of both earnings and revenue multiples, the S&P 500 Software industry's earnings and revenue multiples have expanded considerably over the last ten years, largely the result of the rapid growth of new, pure-cloud entrants that are less profitable but are growing earnings faster than the industry average, a strong recipe for P/E expansion, especially forward P/E shown here.

Net Cash (Debt) Per Share



Source: CFRA, S&P Global Market Intelligence.

- Net cash positions among the constituents of the S&P 500 Software Industry grew steadily from 2010 until 2015, before falling, albeit inconsistently from year to year, to its lowest point in 10 years by the end of 2021.
- The decline in aggregate net cash was due primarily to Oracle's (ORCL) aggressive acceleration of share buybacks, which turned its $9.6 billion net cash position as of May 2018 into $56 billion in net debt as of January 5, 2021, excluding ORCL's pending $28 billion all-cash acquisition of Cerner. Following the close of the Cerner acquisition, ORCL's net debt will exceed $80 billion.
- As of the third quarter of 2022, aggregated net cash position of the S&P 500 Software Industry constituents turned negative, to what is a now a $53 billion net debt position, again largely driven by ORCL. However, we note that if Microsoft's $69 billion Activision Blizzard acquisition, which will be funded from the balance sheet, goes through, the aggregated net debt of the S&P 500 Software Industry constituents will climb over $100 billion.

SOFTWARE MARKET OUTLOOK

RECAP OF 2022

While companies have yet to report results for 4Q22 as of this writing, we expect similar results to what we saw in 3Q22, at least for the top line which saw 13% and 12% Y/Y growth for 2Q22 and 3Q22, respectively, for the 17 constituents of the S&P 500 Software Industry. This is a marked deceleration from the 18% Y/Y growth for 1Q22 and definitely a steeper deceleration than we had expected. This rapid revenue growth deceleration appears to have caught many of the faster-growing cloud-based providers by surprise, with Y/Y EPS growth decelerating from 16% in 1Q22 down to 4% in 3Q22. However, many cloud-based providers have begun to rein in their own spending as they shift from growth maximization to a more balanced operating model with greater optimization for profitability. It usually takes a few quarters to see the full benefit of these streamlining actions, but we started to see some results even in 3Q22 where, despite *revenue* decelerating from 13% in 2Q22 to 12% in 3Q22, *EPS* growth reaccelerated to 8% in 3Q22 from 4% in 2Q22 – a solid demonstration of the inherent operating leverage in the cloud model even when the economic conditions for revenue growth are muted.

Nevertheless, long term, we continue to be quite positive on the software industry, even on the top line, as well as the fundamentals of many – though not all – of its constituents, despite the sell-off in the equity markets in 2022 that has punished the higher multiple stocks of faster-growing companies, many of which are found in the software industry. At the beginning of the year, we were primarily concerned with a resurgent Covid-19 and its Omicron variant. Yet, outside of China, we did not see a return to lockdowns, retail closings, and the other massive disruptions to business operations. However, the strict lockdowns in China, which were as much an exercise in top-down state control as an action in the service of public health, have had a considerably negative impact on the global economy, exacerbated by Russia's invasion of Ukraine. While the Chinese government appears to have relaxed its strict Covid-19 policy, it has lagged in vaccinations, leaving the fundamental problem intact – a large, dense, largely urban population still vulnerable to Covid-19 surges.

In 1Q22, congested supply chain added two more negative developments for the global economy: 1) Putin's invasion of Ukraine in February and its inflationary impact on oil and food prices, not to mention Russia's "crazy talk" about using tactical nuclear weapons in the conflict, making investors more cautious overall and boosting the preference for cash; and 2) the inevitable economic perturbations from the pandemic-induced economic stimulus, the inflationary price we are paying for it, and the resulting rate hikes attempting to curb this inflation. Regardless of whether one thinks things would have been worse without all of that stimulus or not, the reality is the worst inflation we have seen in the U.S. since the 1970s and the highest interest rates in over a decade, likely to go higher.

Obviously, these developments have created much greater uncertainty among investors and a market swinging wildly back and forth, often driven by "macro news", with each swing hinging on daily incremental changes in the global economic and geopolitical landscapes. To be clear, much of the decline in equity prices reflects a genuine global economic downturn that, while it may not entail quite the sharpness and the sense of panic of the 2008-2009 financial crisis, may exceed it in terms of severity and duration.

However, we believe that the degree of this decline in equity prices and the ongoing market volatility are being exacerbated by aspects of market dynamics that have illuminated by the emergent field of behavioral finance over the past few decades. In other words, macro fear now dominates the equity investor psyche. This fear has led to a sell-off in all high multiple stocks regardless of the fundamentals of the underlying companies and regardless of the ability of those companies to adapt or even thrive under these harsher economic conditions.

OUTLOOK FOR 2023

In these times of duress, we redouble our research efforts on the fundamentals to ensure our assumptions are still valid. While this economic downturn is certain to clip topline growth rates, the fundamentals of most – though certainly not all -- software companies remain quite strong. For those following our research, it should not be a surprise that we continue to believe in the adaptability of pure cloud-based providers (e.g., Salesforce.com, ServiceNow, and DataDog), the “tethered cloud” providers of more processor-intensive and latency-sensitive applications (e.g., Adobe and Autodesk), and those traditional software companies where cloud revenue has crossed or is close to crossing 50% of total software revenue (e.g., Microsoft, NICE, and Atlassian).

However, we also very much like the prospects of certain software companies where the processor intensity and latency sensitivity of the applications are so great that cloud delivery is not yet practical but where the ongoing cost-constant increases in computing capacity are now leading to much greater improvements in the customer’s return-on-investment for these applications and the beginning of an explosion in the number and variety of use cases. These are companies we refer to as being on the right side of Moore’s Law. Ansys, with its simulation software used in a variety of high-value engineering and testing use cases, is one of the best examples here. Synopsys and Cadence Design Systems, the two largest suppliers of chip design and testing software, are also very much on the right side of Moore’s Law and, in an interesting case of a tremendously value-adding feedback loop, they are both enabled by and enable Moore’s Law.

It should also come as no surprise that we anticipate rapidly worsening prospects for those software companies where traditional license and, especially, maintenance revenue still account for more than 60% of total software revenue. Oracle, Citrix, Constellation Software, and OpenText are all heavily dependent on revenue from maintenance and support, typically one-year renewable contracts that are vulnerable to more rapid cloud migration, which is likely to occur, in our view, in response to the financial pressure from a worsening economic situation in the second half of 2022 and the first half of 2023. These are times when new paradigms unfold more rapidly in response to economic pressures and the need for greater operational agility and when we see surges in ongoing shifts of market share from the legacy incumbents to newer entrants.

We also see an advantage for the cloud providers in terms of profitability during times of greater economic pressure, because, compared to traditional software providers, cloud app providers, even large ones like Salesforce.com (CRM) and ServiceNow (NOW), can react more quickly than software companies following a more traditional model, cutting expenses and investments focused on sales growth to realize more of their intrinsic operating leverage and expand profitability. Essentially, the cloud model, with its plummeting marginal cost of app delivery and more predictable revenue, is well adapted to sharp economic downturns and can effectively enter "profit maximizing mode" that we alluded to earlier. Further, when the downturn bottoms, cloud app providers can often return to "growth-maximizing mode" just as quickly. Salesforce.com has demonstrated this nimble responsiveness to business conditions multiple times, actually thriving both just after the 2008 financial crisis and following the pandemic in the first half of 2020, taking more market share and accelerating the adoption of cloud computing.

That brings us to another reason why we are especially positive on software in 2022 and 2023, again, particularly on cloud app providers: software is the solution to the problems businesses face when operating under difficult circumstances and during economic downturns. If the software industry’s rebound in the second half of 2020 and 2021 are any indication, times of crisis will delay certain projects and curtail new spending as CFOs assess the situation. Yet, as the smoke clears in subsequent quarters, these crises act to shove forward ongoing software trends like cloud migration and catalyze newer ones such as industrial Internet-of-Things (IoT) solutions that boost manufacturing output and efficiency.

TOP 20 SOFTWARE COMPANIES BY REVENUE			
RANK	COMPANY	TICKER	SOFTWARE REVENUE* *($, in billions)*
1	Microsoft	MSFT	203.1
2	Amazon	AMZN	76.5
3	IBM	IBM	47.1
4	Oracle	ORCL	46.1
5	SAP	SAP	32.5
6	Salesforce	CRM	30.3
7	Alphabet	GOOGL	24.5
8	Adobe	ADBE	17.6
9	Intuit	INTU	13.3
10	VMware	VMW	13.2
11	ServiceNow	NOW	6.9
12	Constellation Software	CSU	6.2
13	Roper Technologies	ROP	6.1
14	Workday	WDAY	5.9
15	Dassault Systèmes	DSY	5.8
16	Synopsys	SNPS	5.1
17	Autodesk	ADSK	4.6
18	Amdocs	DOX	4.6
19	Zoom Video	ZM	4.3
20	Fortinet	FTNT	4.1

*As of CY2021

Source: CFRA, S&P Global Market Intelligence

M&A Environment

TOP M&A DEALS IN 2022

Microsoft buying Activision Blizzard for $68 Billion – Pending
In the largest deal of 2022 and also Microsoft's (MSFT) largest acquisition ever, MSFT announced in January its intention to buy Activision Blizzard (ATVI), which has been dealing with internal turmoil since the state of California brought a civil suit against the company for allegedly perpetuating a culture of sexual discrimination, allegations the company's management did not handle well at all in the summer of 2021.

The deal received considerable scrutiny from the Federal Trade Commission (FTC), which ended up filing suit to block it outright in December. The head of the FTC, Lina Kahn, by her own admission, has sights set on "Big Tech", although most of her attention used to be directed toward Alphabet, Meta, Amazon, and Apple. We learned from a preliminary hearing in early January 2023 that MSFT and the FTC have conducted no settlement negotiations yet and that MSFT is planning on making concessions to the European Union and the U.K., settling with them, and then presenting those terms to the FTC – talk about putting all your cards face up on the table…

Given the concessions to which MSFT may have to submit and the regulatory actions that may be triggered, the risk that this deal falls through is greater than usual, in our view. This is reflected in one of the widest risk arbitrage spreads for a large acquisition in recent memory: as of this writing in early January 2023, ATVI trades at $77 per share despite MSFT's offer at $95 per share.

If it does, MSFT must pay ATVI a $3 billion break-up fee, which would be materially quite positive for ATVI but negligible for MSFT given its $27 billion in net cash and ability to generate $15 billion-$20 billion in free cash flow in a given quarter.

Broadcom buying Vmware for $61 Billion – Pending
In May, Broadcom (BRCM) announced its intention to buy Vmware (VMW), the dominant supplier of server virtualization software, especially in the large enterprise market, as well as a range of software tools enabling enterprises to manage large-scale application deployments within their own data centers and to interoperate with cloud platforms like Amazon AWS and Microsoft Azure at the infrastructural level. According to the deal terms, VMW shareholders will receive ~50% of their compensation in BRCM shares and ~50% in cash, which BRCM is financing with a $32 billion debt issuance. Both Michael Dell, who owns 40.2% of VMW, and Silver Lake Partners, which owns 10%, have agreed to the deal.

Adobe Buying Figma for $19 Billion
Founded in 2012, Figma provides a purely cloud-based design application that has recently become popular among teams engaged in collaborative, multi-person design projects. Figma is expected to report ~$400 million in annualized recurring revenue for 4Q22. If it closes, this acquisition will be Adobe's largest ever. However, in a sign of this FTC's stricter approach to regulation, the deal is now officially under investigation as a potential anti-trust violation given Adobe's "monopolistic control" over the market for design software, which it dominates with its Creative Suite of design applications that includes Photoshop and Illustrator, the de facto standards for designers in digital photo editing and drawing/visual design, respectively.

Citrix to be Bought by Private Equity Firms Vista Partners and Evergreen Coast Capital for $16.5 Billion
Announced in January 2022 before a steep decline in share prices especially of software companies, the plan is for Citrix to be combined with Tibco, another legacy software company owned by Vista. The idea is that the combined company can pool resources to make the transition to a shared cloud-based platform more economical. The deal was completed on September 30, 2022.

Kaseya Buying Datto for $6.2 Billion – Closed in June
Kaseya is a privately held provider of apps and solutions to help manage and secure complex IT operations as well as provide back-up and restoration of high value data. Datto provides security and other solutions specifically targeted at managed IT service providers.

Alphabet Acquiring Mandiant for $5.4 Billion
Announced in March, this is Alphabet's (GOOGL) second largest acquisition ever and is focused on adding Mandiant's (MNDT) cloud-based security software to its Google Cloud infrastructure and platform offerings. MDNT has emerged as a leader in software and consulting services for cyberattack detection and counter measures. In 2020, the company launched a cloud-based cybersecurity application that is the focus of Google Cloud's interest in the company. Any large acquisition by GOOGL is likely to attract more regulatory scrutiny than acquirers outside of "Big Tech", and, in April 2022, the U.S. Department of Justice requested additional information, an obvious sign of such a higher degree of scrutiny. However, given that this acquisition is focused on bolstering Google Cloud, which trails both Azure and AWS by a significant margin (in 2021, AWS revenue was $62.2 billion, Azure revenue was roughly $35 billion, and Google Cloud was $5.4 billion), this deal likely has a good chance of closing, in our view, although the U.S. government may very well want some additional concessions from GOOGL first.

CFRA's "Four Key Trends in Enterprise Software"

In August 2020, we introduced what we simply call "Four Key Trends in Enterprise Software", which we believe are critical to understand in order to invest successfully in the software industry. These trends have been building for over a decade and will continue to play out over the next decade and, in the case of AI, well beyond.

Ongoing Migration of Enterprise Applications to the Cloud – still < 50% complete
There are three main categories of cloud services: Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS). These definitions have remained constant for about a decade, although the lines are starting to blur, especially between IaaS and PaaS services. Or, more accurately, for the three large dominant providers of both service types – Amazon AWS, Google Cloud – and it is becoming more difficult to distinguish between revenue generated from IaaS and revenue generated from PaaS given bundling deals and the sheer number of additional services applicable to either.

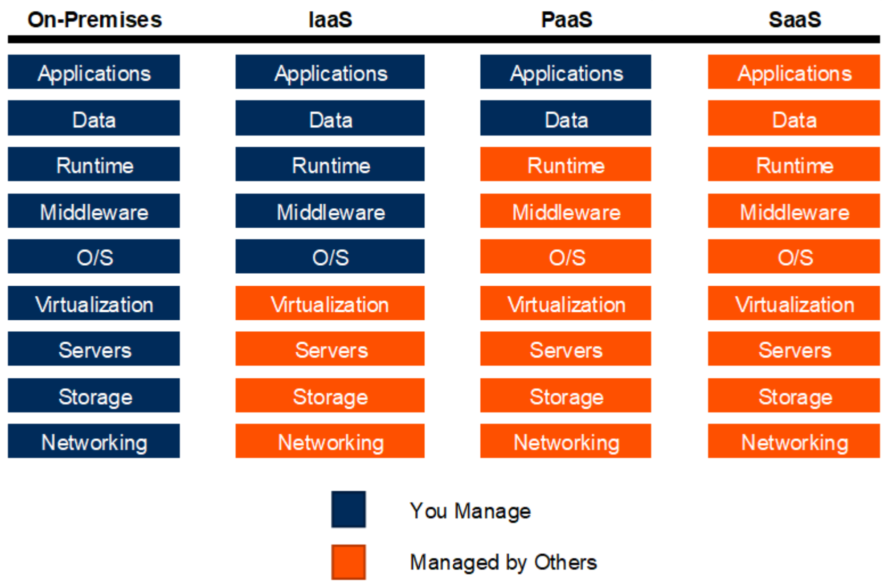
◆ **Software-as-a-Service (SaaS)** is the "original" cloud service where customers simply use a standard browser to access a web-based application operated by the SaaS provider. Salesforce.com, Workday, ServiceNow, and NetSuite (now Oracle) are among the earliest SaaS pioneers and continue to operate some the largest SaaS offerings in terms of number of subscribers, revenue, and/or data volume. If one wanted to maintain consistent definitions of these services, SaaS should really be called Application-as-a-Service.

◆ **Infrastructure-as-a-Service (IaaS)** gives customers access to virtual server and storage instances over the internet, enabling the customers to develop and run any application just as if it were running in their own data center. Initially, IaaS growth was driven by testing and development, which occupies about 15% of the IT infrastructure resources of a typical large enterprise on average but with resource requirements that fluctuate wildly even within a single day. As AWS matured and began offering guarantees for various levels of performance for multiple metrics, enterprises began running full-fledged apps on the service, which was quickly followed by start-up SaaS providers running their cloud applications to AWS. After 2015, more and more existing SaaS providers began migrating their applications to AWS, and increasingly, to Microsoft Azure and Google Cloud as well, abandoning operations of their own data centers and cloud infrastructures.

◆ **Platform-as-a-Service (PaaS)** offerings occupy a middle ground between a full-fledged app that can be used “out-of-the-box” and a “raw server and storage” instance making the customer responsible for installing and configuring its own “full stack” including database software, if needed, all the way down to the operating system. The advantage of PaaS is that it provides everything required to develop custom applications for 90% enterprise use cases without burdening developers with infrastructural minutia that no longer adds value for most enterprises.

However, generally, PaaS is not optimal for migrating legacy applications “as is” to the cloud. There are two big exceptions to this: apps originally developed within the Microsoft .Net framework are more easily moved to Azure PaaS services and custom apps developed on an Oracle database are more easily moved to Oracle's Autonomous Database PaaS service. Also, because they present all the middle layers of the stack to the developer as a “black box”, PaaS offerings tend to be less standardized, more vendor-specific, and prone to lock-in. In addition to the big three cloud providers, significant PaaS offerings include Salesforce.com’s Lightning (also known as Force.com), Oracle Cloud, Oracle Autonomous Database, IBM Cloud, IBM's RedHat OpenShift, and growing variety of more functionally targeted platforms ServiceNow's Now Platform, New Relic's New Relic One, Twilio, and Manhattan Associates' Manhattan One are just a few examples. And, yes, the software industry is not known for its branding creativity.

DIFFERENCES OF CLOUD COMPUTING SERVICE MODELS



Source: Microsoft

Prior to the 2008 financial crisis, the Phase One stage of the migration to cloud computing was dominated by small (<100 employees) and mid-sized (<1,000 employees) businesses. Toward the end of that period, individuals and even some departments within large enterprises began to use their web browsers and credit cards to subscribe to certain Software-as-a-Service (SaaS) applications like CRM from the likes of Salesforce.com, RightNow (now part of Oracle), or selected functions within HCM from companies like SuccessFactors (employee evaluation tracking, now a part of SAP) and Concur (expense tracking/report generation, also now a part of SAP) or trouble ticket tracking/management within the IT systems management category (e.g., ServiceNow). Also, in the later portion of this period, software developers began using Amazon's AWS service for development and testing, since they could have access to virtual servers and cloud storage within minutes rather than having to wait for days or weeks for IT to install a new physical server in the data center and make it available for development and testing.

The second phase of cloud migration began during the recovery from the financial crisis and was seen among many in the C-suite of Forbes Global 2000 companies as a cost-reducing exercise in preparation for future crises. With the seeds planted among many large enterprise sales and marketing employees, as well as within IT departments themselves, SaaS version of CRM applications, IT service management/ITSM, and an increasing number of HCM applications began replacing their client server equivalents that made up on-premise installations. In addition, IaaS and PaaS offerings began to expand well beyond their test-dev use case, and an increasing number of in-production applications began to move to services like Amazon's AWS – including the cloud software of other SaaS providers. At this point, most new SaaS start-ups run their application on top of AWS or similar service.

The current phase is characterized by some of the largest and most conservative enterprises moving their most mission-critical applications to the cloud, such as those falling into the Enterprise Resource Planning (ERP) category of apps, especially the financial modules within ERP, and – even more critically – the general ledger accounting module. In fact, all of the 753 organizations surveyed in Flexera's 2022 State of the Cloud report (latest available) are using at least one public or private cloud, with 96% utilizing at least one public cloud, 84% having at least one private cloud, and 80% using hybrid cloud.

We see cloud-based software businesses exhibiting a greater degree of operating leverage and – most critically – for a longer duration than many investors expect. Traditional software companies that sell licenses, upgrades, and maintenance contracts for software installed and operated on the customer premises (i.e., “on-prem”) typically exhibit a substantial degree of operational leverage over a fairly short period of time. This is particularly true of enterprise software companies, especially those focused on more complex, big-ticket solutions with fairly high barriers to entry, high switching costs, and pricing power. Yet, operating margins for even the most profitable of the traditional software companies typically level off in the mid-thirties and rarely exceed 40%. In contrast, cloud-based software companies take longer to break even, reaping operational and scale efficiencies more slowly than traditional counterparts.

However, we see indications that cloud-based software vendors, whether they are “cloud-first” like Salesforce.com and ServiceNow or traditional software suppliers in a transition to cloud, do not have such a ceiling and could plausibly break through 50% or even 60% operating margins in the long term. For example, Visa, which we see as a relatively mature provider of “transaction processing as a service” software in the cloud (before anyone even coined the term “cloud”), consistently generates an operating margin above 60%.

The Rise of Meta-Software – Software that’s “about” software, a more investable way to play digital transformation

We define meta-software as software that automates or aids in the creation, testing, deployment, operation, performance, and security of software applications themselves. Meta-software sub-segments generally fall under one of three overall software segment categories: software development tools, IT operations software, and IT security software. We see two key drivers of meta-software overall:

1) the continued migration from client-server and web-based application architectures to a micro-services-based architecture (MSA), requiring an upgrade to the more powerful software tools optimized for the dramatic increase in complexity that comes with MSA; and

2) the shifting role of enterprise software from a tool for managing the means of production to actually becoming the means of production itself. In our view, as software becomes increasingly mission critical – to the point where the means of production can no longer be switched back to “manual” – companies will correspondingly require a lot more and a lot better software to help manage and automate the secure production and operation of the software that is now the means of production. For Uber, Expedia, Airbnb, any online brokerages, and a growing number of companies across industries, software is the means of production. If it fails, the business stops cold. There is no "man-config" mode and no way to put the software genie back in the bottle. Further, as a software business scales, it becomes very difficult if not impossible for humans to handle that software and therefore that business without the help of a lot of equally scalable software. In a way, the rise of meta-software can be seen a derivative of digital transformation and, in our view, is actually a more investable way "to play" the digital transformation trend.

The “Cognification” of Software – applications you do not have to program or configure but rather train on data

We have no doubt that software will be increasingly cognified, which means that code will be able to act on our behalf using increasingly softer and fuzzier logic until we recognize that it is, at least within a well-defined context in which it operates – e.g., recognizing a face, diagnosing cancer, optimizing a supply chain, hedging currency risk, prioritizing sales leads, etc. – using its “best judgement.”

We specifically use the phrase “cognification of software” as it embodies three elements we think are necessary for commercial success, especially in the enterprise application market:

◆ **Incremental**. The incremental augmentation of software code that is already in production because that is where the data is. All forms of cognified software gain their cognitive ability by “training” on data. With greater volume, faster changing, and more up-to-date and more relevant data, software will learn to make better judgment calls.

◆ **Bounded**. The cognitive capacity is targeted at a well-defined range of decisions within a bounded context. By bounding the context of the decision that the software will make (e.g., assisting a bank customer through a transfer of funds versus answering questions about the history of philosophy), accuracy and return on investment increase exponentially.

◆ **Agile**. There is enough architectural flexibility in the software so that it can be quickly improved iteratively using agile, dev-ops-oriented processes, with each "increment of cognition" upping the ability to act autonomously and converging to that point where it appears that it is using "its best judgment." Of course, for in-production software, the flexibility that comes with the implementation of Agile DevOps processes must be accompanied by a similar degree of resilience and security.

As is often the case with implementing a new technology, adding an increment of cognition will initially seem to be more trouble than it is worth, akin to training a new employee to perform some semi-routinized tasks that have just enough "exceptions to the rule" to make it difficult to let them go and learn "autonomously;" it just seems much easier to do it yourself. However, once sufficiently trained, even mildly cognified software operating in a limited scope will, we forecast, add tremendous value, especially for tasks that humans consider boring (e.g., checking luggage in an airport X-ray machine), cognitively stressful for extended periods with limited upside reward (e.g., air traffic control), or emotionally unpleasant (e.g., negotiating a divorce).

Digital Transformation: software substituting for labor and capital in the means of production in virtually every industry

Digital Transformation is one of the most popular terms these days in both technology and business realms, being the subject of countless papers, books, business articles, blog posts, MBA courses, and a range of marketing campaigns. Yet, it is also one of the most ill-defined terms in both technology and business realms. We define digital transformation as the substitution of software for labor and capital in the means of production. This contrasts with applications employed to help improve and automate the *management* of the means of production – Enterprise Resource Management (ERP), Supply Chain Management (SCM), Human Capital Management (HCM), Customer Relationship Management (CRM), and Business Intelligence (BI). ERP, SCM, HCM, and CRM have been around since at least the 1980s, if not before, while BI emerged a little later in the late 90s.

Obviously, there are some cases where the distinction between management and the means of production can be fuzzy. Arguably, the category of enterprise software that blurs the distinction most is Product Lifecycle Management (PLM), a relatively new category of enterprise applications, having gained traction mostly just in the past decade. PLM straddles both the means of production and management in terms of its primary users and in terms of the software with which it integrates. On the management side, PLM touches or is integrated with ERP modules (especially accounting) or even SCM, and CRM applications. PLM is often integrated with engineering and design applications (e.g., from the likes of Autodesk and PTC) that are unambiguously employed in the means of production, especially for high value products with intensive design cycles and long lives "in the field" (e.g., architecture/construction, aerospace, and auto industries).

Later in this report, we cover the details of what digital transformation is by delineating what it is not and by putting the term in the historical context of the information technology history. Please see *Rule #6. "What Digital Transformation Actually Means"* under the section, *The First Eight Rules of Investing in Software-based Businesses*.

EIGHT “RULES” FOR UNDERSTANDING AND INVESTING IN THE SOFTWARE INDUSTRY

RULE #1. NO RULES!

Stop following, reading, or even glancing at lists of rules. Emerging from the listicle (a common form of online expression where an article is structured as a numbered list to hold readers' attention), the rule list has become the most insidiously addictive, scroll-inducing click-bait on the internet. There are so many lists of rules: 32 ways to improve your TikTok profile, destroying democracy in 8 easy steps, 15 rules of surviving a zombie apocalypse, 10 rules of dating, 8 rules to a successful marriage, The 48 Rules of Power (seems like way too many rules), 10 “golden rules” of management (What? No bronze rules?), the 3 golden rules of accounting (seems like accounting would have more rules), the 3 rules of real estate investing (This one is a total rip-off! How is “location” even a rule?), ad infinitum.

Lists of rules work by exploiting our deep desire to algorithmicize our lives and choices. If we can just identify the right rules, call it “Rule Set X,” and, through sheer determination and repetition, master its tasks, check all of its boxes, and hit all of its Key Performance Indicators (KPI), then all of our dreams will come true. If we fail, our failure belongs only to us, transparently documented, with no wiggle room for low-bar benchmarks, puffery, organizational politics, or innuendo. Lists of rules appear to give us a path to meritocratic utopia. Once we achieve mastery of “Rule Set X,” then coach, client, followers, fans, or the stock market yells, “GREAT JOB!” Then, we get a dopamine squirt for reinforcement, pick up and enjoy our awarded “utils,” and repeat. There’s even a rule set so powerful, the diligent follower is promised “the world and everything that’s in it.”

Subversive Recursion

Unfortunately, but actually fortunately, nothing worthwhile in life is attained by only following rules. That includes mastering the craft of discovering, analyzing, and valuing the equities of software-based businesses. The problem is that rules are not exogenous. Over time, rules respond and evolve while you follow them and so does the market. As participants join in to follow any rule that produces a profit, their actions steadily dislodge the correlations between the rule and market behavior, eventually sapping the rule of any predictive value – if it ever even had any to begin with.

The irony is that, if you get really good at following rules, eventually you become really bad at picking stocks. How fast that happens depends on the pace of change, which has never seemed faster for all human endeavors than it is today, but especially for software-based businesses. So, as comforting as a “time-tested” set of rules may seem, the term begs the ironic question, "how up-to-date is that time test?"

> *“The future doesn’t just happen. It’s not etched or written or coded anywhere. There’s no algorithm or formula that says technology will do X, so Y is sure to happen.”*
> **– Eric Schmidt, Former CEO of Alphabet/Google**

Looser than Dalio's Principles, Tighter than the Tao De Ching

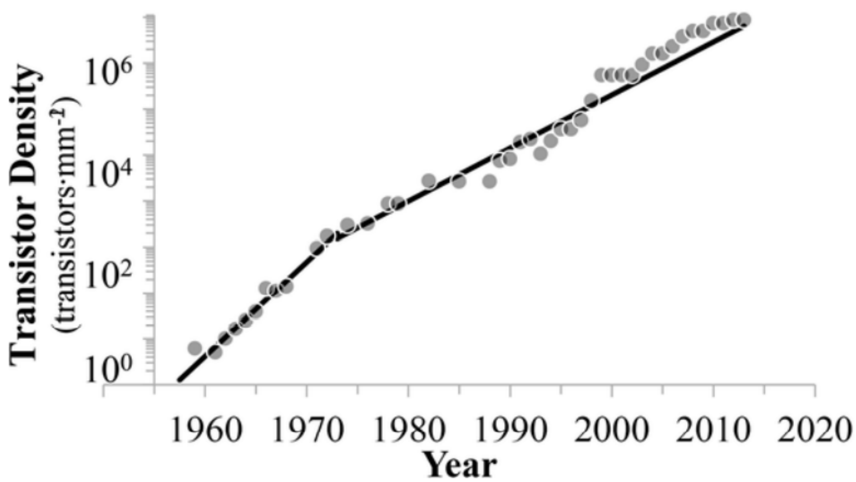
At best, these “rules” are frameworks, mental models, modes of thinking, open-source thought chunks, little booster hits of cognitive nitro, engaging narratives to prompt helpful questions, or metaphors to help understand the abstraction inherent in software and software-based businesses. Maybe, these are just eight descriptions of historical patterns that you cannot rely on to repeat but that might rhyme, making the next lyric a bit more predictable than a random walk. At worst, they are eight things to think about when you are trying to pick stocks, while also following someone else's rules.

RULE #2. MOORE'S LAW RULES!

We think it's crucially important to understand and be continually mindful of the following: all of the tremendous innovation and progress in software, the growth in software industry revenue, and the creation of all software-based businesses were fed by a cost-constant growth in transistor density – a rough proxy for computing capacity per unit area per unit cost – of more than 40% per year since the first commercial transistor in 1954. That cost-constant growth in computing capacity is delivered to the software industry by the semiconductor industry with remarkable consistency after the first commercial integrated circuit (IC or "chip") in 1964 with 120 transistors, about 1 transistor per mm2, or Tr/mm2 = 1. Today, a chip of similar size features over 15 billion transistors, or Tr/mm2 = 100,000,000.

This rate of improvement in computing capacity has remained so consistent that it seems like a law of nature, but it's really just an observation about human economic behavior coevolving with technological progress. In fact, this predictive observation was famously made by Intel co-founder Gordon Moore in a 1965 Electronics Magazine article stating that, holding cost and surface area constant, the number of transistors manufactured on an IC doubles every 18 months. In 1975, Caltech professor, Carver Mead, coined and popularized the term "Moore's Law," and Moore extended the transistor density doubling pace to 24 months (i.e., biennially). For Intel's x86 CPUs, Moore's Law remained true until 2014 before slowing, as depicted in the graph below.

RATE OF TRANSISTOR DENSITY INCREASE



Source: Burg D, Ausubel JH (2021) Moore's Law revisited through Intel chip density. PLOS ONE 16(8): e0256245.

Below are some examples to help illustrate how foundational this exponential growth in transistor density has been in enabling all of the software and software-based services we enjoy today:

- The first iPhone debuted in 2007; if it had been constructed in 1985 with the transistor density of the logic chips at the time, the mobile processor alone would have covered a football field.
- The computational capacity required to render the velociraptors in the first Jurassic Park movie came from a network of over 100 Silicon Graphics workstations, computing capacity that became available on a single notebook PC by 2005 and on your phone by 2012.
- If one were to construct a data center using chips from 1978 but with the computing and data storage capacity of today's hyperscale data centers from the likes of Amazon or Microsoft, it would be larger than Connecticut – to say nothing of the power consumption!

The astounding growth in cost-constant computing capacity per unit area is foundational to understanding software and to investing in software-based businesses. In our view, software running on computers built from integrated circuits that improve geometrically every year has been critical for every major innovation since the excimer laser, which itself was key in propelling Moore's Law forward in the 1980s and was the last Moore's Law-enabling innovation achieved without the help of software. In other words, the excimer laser was the last Moore's Law-enabling innovation that occurred without help from Moore's Law.

Transistor density is still doubling biennially for chips made by Taiwan Semiconductor (TSMC), which manufactures chips for: 1) fabless chip firms (e.g., Qualcomm, AMD, and NVIDIA); 2) hardware companies that design their own chips built into mobile devices (e.g., Apple) and systems (e.g., Cisco); 3) cloud service providers operating “hyperscale” data centers (e.g., Alphabet, Microsoft, Amazon, and Baidu). However, it appears that for TSMC, cost/transistor is no longer falling at the same rate and is actually flattening.

> *Anything compounding at that rate so often and so consistently is going to have a much bigger impact than anyone expects. Son,* ***the future is going to be VERY different from the past****.* [emphasis added, referring to the rate of cost-constant increases in computing capacity in 1980, a phenomenon that had just been dubbed “Moore’s Law” in 1975]
> **– Charles “Bucky” Freeman, Director of Research at the U.S. Department of Defense’s Night Vision and Electro-Optics Laboratory**

It’s also critical to understand that Moore's Law applies not just to CPUs but all digital processors, memory, and solid-state storage, but with telling variations regarding: 1) lag – how far behind in years and in transistor density a given chip is from the most advanced manufacturing process; and 2) cadence – how often a chip is upgraded to a more advanced process, yielding higher transistor density.

Moore's Law applies to the chips running Ethernet switches and internet routers, to baseband processors responsible for cellular connectivity in mobile phones, to the transceiver used to read/write data from/to hard drives, to digital cameras, to microcontrollers in car engines, to the guidance system of Tomahawk cruise missiles, to the signal processing complex within MRI machines, and to any consumer product we call “smart”. It's all software, even if you can't see or control the app, fed by Moore's Law, that is the ultimate force behind all incremental growth in developed economies since 1980.

> *The way Moore's Law occurs in computing is really unprecedented in other walks of life. If the Boeing 747 obeyed Moore's Law, it would travel a million miles an hour, it would be shrunken down in size, and a trip to New York would cost about five dollars.* ***Those enormous changes just aren't part of our everyday experience.*** *[emphasis added]*
> **– Nathan Myhrvold, former Microsoft CTO and co-founder of Intellectual Ventures**

RULE #3. THE SECOND HALF OF THE CHESSBOARD

Here, we are going to tell you a story to help emote the upside potential of software. The allegory of the emperor of India and the inventor of chess has been used several times over the years to impart an understanding, or rather, an emotional appreciation of the impact of regular geometric improvement over time. Among those employing this allegorical comparison of the legend of the chessboard directly with Moore's Law, Erik Brynjolfsson and Andrew McAfee were some of the first to do it in their book, *Race Against the Machine*.

Here's our take:

According to legend, the emperor of India was so impressed by the game of chess, he granted the game's inventor any reward she wished. The woman replied, "I'm a simple woman. Give me one grain of rice for the first square of the chess board, two grains for the next square, four for the next, and so on for all 64 squares, with each square having double the number of grains as the square before." The emperor agreed, amazed that the inventor asked for such a small reward, or so he thought…

THE LEGEND OF THE EMPEROR OF INDIA AND THE INVENTOR OF CHESS

						64	128	One handful of rice
256	512	1,024	2,048	4,096	8,192	16K	32K	1kg bag of rice
66K	131K	262K	524K	1M	2M	4M	8M	2 wagons full of rice
17M	34M	67M	134M	268M	537M	1B	2B	One harvest for a large farm
4B	9B	17B	34B	69B	137B	275B	550B	One harvest for a district
1T	2T	4T	9T	18T	35T	70T	141T	One harvest for all of India
281T	563T	1aa	2aa	5aa	9aa	18aa	36aa	20,000 container ships full of rice
72aa	144aa	288aa	576aa	1bb	2bb	5bb	9bb	A rice pile larger than Mt. Everest

Another perspective, the 64th square holds so much rice that, placed end-to-end, all of the grains would stretch from Earth to our nearest star, Alpha Centuri, and back!

Source: CFRA.

On the 16th square, the emperor is not worried as this is only about 1kg bag of rice. On the 24th square, he is still not concerned as the amount of rice fits into two wagons. However, on the 32nd square, with an amount of rice now equal to a large harvest from a farm, the emperor realizes the magnitude of his mistake. Somewhere after entering the 2nd half of the chess board, the emperor calculates that, on the 64th square, he owes the inventor a rice pile bigger than Mt. Everest.

In one version of the story, the emperor becomes so enraged when he realizes he's been conned that he has the inventor beheaded. In our version, the emperor is too stubborn to realize what's going on, so the inventor yanks him off his throne and takes him out with a kimura, her signature Brazilian jiu-jitsu move. She dons his clothes, puts her wig on his head, assumes the throne, finally ends the patriarchy, and *ushers in a new world order, where, in order to graduate high school, all children must pass calculus and statistics taught as both mathematics and poetry. And they all lived unfettered ever after.*

Stop Being So Surprised

Like the seemingly “normal” progression of rice volumes on the first half of the chessboard, some computational tasks seem impossible, out of reach, highly improbable, or just totally unfeasible for a long time… Then, all of a sudden, they are not. Compounding geometric growth in computing capacity crosses thresholds for previously unfeasible things and shocks us over and over again. Soon after a threshold is crossed, it often becomes computationally trivial, cheap, and ubiquitous – and that shocks us too. Many people, institutional investors included, are continuously caught by surprise by the derivative impact of what we all agree is an exponential phenomenon that, while difficult to grasp, can be extrapolated into the future.

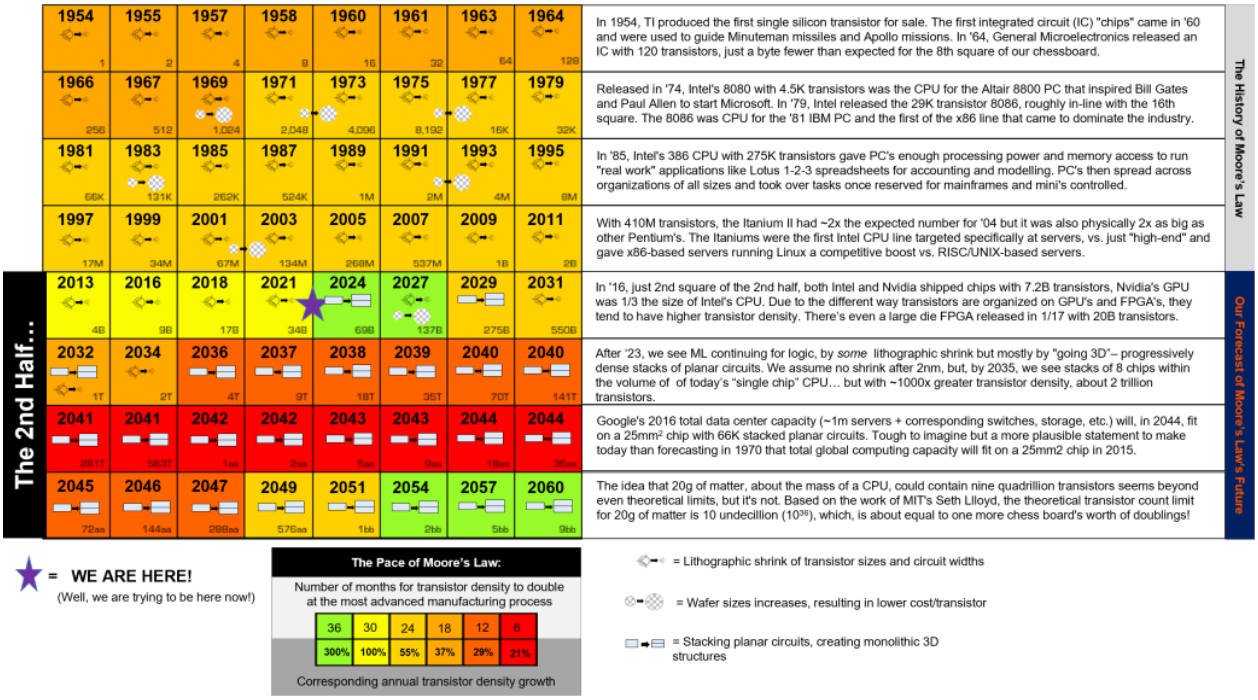
This allegory holds two crucial implications for investing in software-based businesses fueled by Moore's Law:

1. Humans are quite poor at grasping the progression and ultimate impact of non-linear phenomena.
2. “The second half of the chess board” is where the magnitudes surprise us and things get really interesting.

Brynjolfsson and McAfee remarked that if the doubling of computing capacity driven by Moore's Law were mapped onto a chessboard like the grains of rice promised to the inventor of chess, with the year of the first commercially available transistor, 1954, being square one, then we just entered the second half of the chess board in 2013. Obviously, the second half of the chessboard is a metaphor, yet it neatly summarizes our conviction in the tremendous upside potential ahead. In addition, it encapsulates that among all public equities, those of software-based businesses fueled by Moore's Law hold the greatest potential by far to outperform the overall equity market, with their long-term potential being much greater than any single sector.

The ability to shrink the size of transistors and their interconnects has been the primary driver of Moore's Law to date, both in terms of a chip's transistor density and its cost. Increasing the size of the silicon wafers from which chips are fabricated, thus increasing the number of chips that can be manufactured from each wafer, has also played a major role in reducing chip cost. On the chessboard diagram below, we map the history of Moore’s Law with each square representing a transistor density doubling and the year in which it took place. Then, we forecast Moore’s Law through 2060 based on both shrink and “going 3D,” i.e., stacking of chips with increasing vertical density. This is already being done for special chip packages (e.g., for servers), but we see it going increasingly mainstream at the leading edge in 2024. Below, we have mapped out transistor density doublings to-date on a chessboard and then forecast transistor density growth through the “end of the chessboard”.

MOORE'S LAW AND THE SECOND HALF OF THE CHESSBOARD



Source: CFRA.

Wait, isn't Moore's Law slowing, stopping, dying, or dead? There is a humorous corollary to Moore's Law: the number of people predicting the end of Moore's Law doubles every two years. Moore's Rate is not declining, though it has for Intel, which had been the first in the semiconductor industry to fabricate digital logic chips at the leading edge from the 1980s until about 2017, when TSMC took Intel's place as "Moore's vanguard", the first among chip makers to stamp out chips at the next jump up in Tr/mm^2. However, TSMC has not been able to maintain the necessary 50% reduction in cost per transistor and therefore, its doubling of transistor density has not been cost constant, although we have yet to see a commensurate fall in demand.

But even if the semiconductor industry slows way down and achieves a "Moore's Rate" of "just" 21% annual growth in transistor density (aka computing capacity) per unit cost each year, that would result in a doubling every three years. Such growth would continue to enable software to break one computational barrier and therefore "value barrier" after another. Software will continue to solve previously intractable problems, make recently solved problems trivial and cheaper to solve, and create completely new opportunities most had not even thought about.

Yet, what if, as we have forecasted in the chessboard diagram above, Moore's Rate accelerates as transistor density is derived less from horizontal shrink and more from progressively denser vertical stacking, in effect, building ICs with increasingly higher transistor densities per unit cost in 3D – i.e., $Tr/mm^3/\$$ – not just the density of transistors on a chip's 2D surface area. Increasingly dense packages of stacked chips are already being manufactured, though the cost limits the feasibility to certain applications, such as servers in hyperscale data centers that can use all of that $Tr/mm^3/\$$.

Positively Recursive to the Max

However, the cost of stacking and integrating chips together in a single package will certainly fall with greater volume and with design and process innovations. These innovations will not spring from Athena's head or anyone else's unless aided by software – software made faster and more powerful by Moore's

Law. Yes, for many decades now, the reason for Moore's Law has been Moore's Law, perhaps the most powerfully positive of the recursive loops that have driven human technological and economic progress – as if there's much difference between the two – and certainly the fastest acting.

RULE #4. WHAT INNOVATION MEANS TODAY

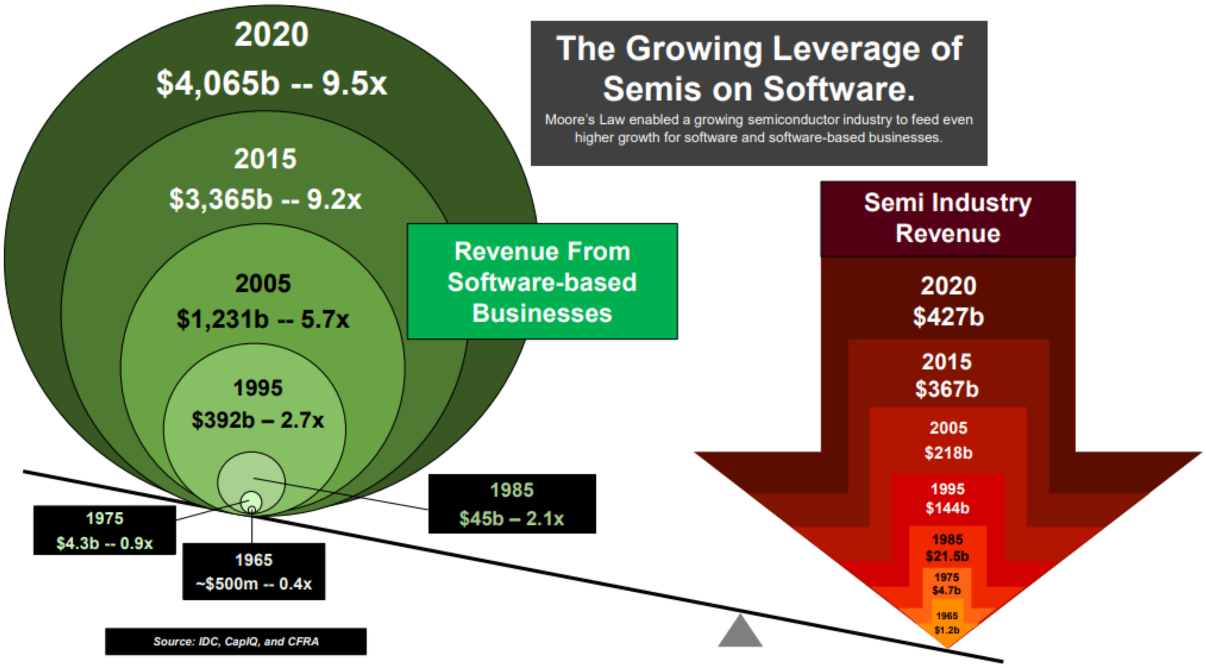
Here is where we start to think about what software could do, what it can be, and how the software industry would change given a future state of Tr/mm^3/$ – i.e., transistor density computing capacity per unit cost. Another way to put it is: what things could software accomplish in the future if only it could economically access some amount of computing capacity within a particular form factor? That form factor could fit in a pocket, a backpack, the nosecone of an ICBM (Inter-Continental Ballistic Missile), on a desktop, within a very well airconditioned building the size of a football field, or in the space that a surgeon created behind someone's eyeball near their optic nerve.

Our first point here is that, for every major innovation we can think of since 1980, software's role has been indispensable. Even if a few innovations may have been stumbled upon and made economically viable without increasingly powerful software, for most, it is quite unlikely. When one studies the details of any innovation over the last 40 years, it is the combinations of human-based and software-based cognition or, if one prefers, human cognition aided by software that has produced all of the major innovations, even if the result contained no software at all – from better all-weather tires to Covid-19 vaccines. At this stage, it's getting hard to extricate software from virtually every human endeavor, and that's kind of our point.

> *Information technology and business are becoming inextricably interwoven. I don't think anybody can talk meaningfully about one without the talking about the other.*
> **– Bill Gates, co-founder and former CEO of Microsoft**

So, for us, innovations are combinations of human- and software-based cognition made possible by some recent state of Tr/mm^3/$ at the leading edge and $/Tr at the trailing edge. While this may seem deterministic, even formulaic, it is far from it. The universe of possible innovations grows right along with increases in Tr/mm^3/$ at the leading edge and $/Tr at the trailing edge and is already massive – as we have stated previously, touching virtually every major human endeavor. In a way, the impact of the entire semiconductor industry is multiplied through the software industry.

Or, in other words, because of Moore's Law, as the semiconductor industry delivers more Tr/mm^3/$ at the leading edge and lower $/Tr at the trailing edge, the software industry generates increasing value and reaps increasing returns. That rising leverage can be plainly seen in the simple ratio of software industry revenue to semiconductor industry revenue in the diagram below.



In our "Right Side/Wrong Side of Moore's Law" Construct, we say a company is on the "Right Side of Moore's Law" to the degree with which it benefits from the leverage depicted above. More specifically, a company is on the right side of Moore's Law to the degree with which its innovation – some combination of human- and software-based cognition – optimally exploits the present and imminent states of Tr/mm^3/$ at the leading edge and $/Tr at the trailing edge to generate value, as well as its ability to capture as much of that value as possible. A company remains on the right side of Moore's Law to the degree with which it can iterate, protect, and extract value from that innovation in subsequent states of Tr/mm^3/$ at the leading edge and $/Tr at the trailing edge. This is accomplished by enhancing the software with new features, increasing its scalability, improving performance, making it easier to use, expanding its compatibility across new hardware systems and with other types of software, and so on.

For a company heavily on the right side of Moore's Law, every added increment of Tr/mm^3/$ from the present state forward is not just a knee in the curve for computing capacity – it is a knee in the curve of the value that company's software can deliver with that added computing capacity. For example, we say Ansys is heavily on the right side of Moore's Law because its simulation software is not only tremendously computationally intensive, but it also produces a high marginal return for each surplus increment of Tr/mm^3/$. Ansys' simulation software can be economically applied to an increasingly wider range of use cases, with much more accurate results, at a lower cost, etc.

However, that was not always the case. Even in the 1990s, the computational requirement for software simulation was so great relative to the state of computational capacity per unit cost that it could only be applied in select, high-value/high-payoff situations with hundreds of millions of dollars on the line, such as in the design and testing of products in the aerospace industry. Being on the right side of Moore's Law does not simply mean computationally intensive software – it must apply at every increment of additional Tr/mm^3/$ from the present state forward.

This study of technological change… revealed two types of technology change, each with very different effects on the industry's leaders. Technologies of the first sort sustained the industry's rate of improvement in product performance… and ranged in difficulty from incremental to radical. The industry's dominant firms always led in developing and adopting these technologies. By contrast, innovations of the second sort disrupted or redefined performance trajectories – and consistently resulted in the failure of the industry's leading firms.
– Clayton Christensen, former Professor at Harvard Business School and author of The Innovator's Dilemma

One of the best ways a company can lock-in a position on the right side of Moore's Law is by turning their app into a platform or by building a platform to support their app along with a variety of other apps, usually including the larger universe of apps from third parties, growing the user base and expanding the overall ecosystem, thus giving the company's software a deeper grip among its customers, especially to the degree with which the platform or entire eco-system helps its customers innovate and stay on the right side of Moore's Law themselves. Companies that establish platforms continually fueled by Moore's Law hold the best chance to parlay their positions across paradigms and eras. They are what we call era-hoppers, and they create the most shareholder value over the long term.

Another way of putting this: going back to the chessboard metaphor, how many chessboard squares can a company string together in row in which its software generates and captures value – preferably progressively greater value – in each square? This is not an easy task and incorporates many more variables than one might suppose at first glance, but it is the crux of fundamental analysis of software-based businesses, in our view. If a company can no longer use subsequent states of Tr/mm^3/$ and $/Tr to add value or if that value declines with each subsequent state of Tr/mm^3/$ and $/Tr, then we term it to be "on the wrong side of Moore's Law." Then, Moore's Law works in reverse to commoditize or make irrelevant the company's existing product or service, effectively sapping or transferring its value to other companies' solutions with each delivery of surplus computing capacity at the leading edge or each cost reduction for a fixed amount of computing capacity at the trailing edge.

Such a thought exercise should be counterbalanced by a company's ability to: 1) generate value from its older innovations that are still being purchased and deployed for the first time by "late mainstream" and "laggard" customer types on the adoption curve; and 2) continue capturing value from the installed base of its older innovations – essentially a measure of "customer stickiness" that includes the cost and risk of switching vendors. Depending on the area of software, this second consideration might entail a bigger, more complex architectural shift for an enterprise, impacting other applications and entailing a bigger decision about software architecture. In turn, this situation often encourages the enterprise to delay smaller changes and adoptions of newer innovations and represents a major source of customer stickiness.

However, this strategy to protect ongoing value-capture has its limits and, eventually, some Tr/mm^3/$ or $/Tr threshold will be crossed that makes the next big architectural shift (i.e., move to client-server, move to web-based computing, move to cloud, etc.) an urgent imperative for the customer. Often, exogenous events (e.g., the recession of 1991, Y2K, the post 9-11 economic downturn, the 2008 financial crisis, or a global pandemic) acts to catalyze such an architectural shift. So, is innovation predictable? Mostly… with caution. Even when one understands all this Moore's Law leverage stuff and can envision future states of leading-edge Tr/mm^3/$ and trailing edge $/Tr, it is still hard to see what novel combinations of human and software cognition will be created and what opportunities they can address.

Monitoring what is being funded at the venture level and the start-ups that are gaining some traction or at least "buzz" certainly helps in that regard. Even if they never make it to IPO, one will at least become familiar with the companies and technologies that public software companies are likely to acquire a few years down the road. Paying attention to predictions and "watch lists" for new technology adoption from industry analysis firms like IDC and Gartner is also helpful, but not gospel.

Nevertheless, even if imperfect and recognized as such, in order to forecast the fundamentals of software-based businesses, it is important to have a sense of what's coming. In order to get that, one must stay up-to-date on the latest developments, but familiarity with software's past is also quite helpful to see how the lyrics of history might rhyme as the industry evolves from here.

The Cyclicality of Software's Evolution

While truly novel innovations are often hard to see coming regardless of one's mental model or helpful metaphor, in software, there are always security, legacy integration, service-level enforcement, and/or consolidated management markets somewhere. As software exploits its surplus computing capacity, generates value, and begins to be more broadly adopted, it invariably creates new problems. Popping up like clockwork, many of these new problems are the same as the old new problems.

Generally speaking, as investors, it is easier to see these "new problem" opportunities earlier than truly novel opportunities because there tends to be a certain cyclicality and repetition to them. For example, let "NewSoft" represent some new software capability that is creating value and gaining traction. NewSoft can be: 1) Small, relatively limited in scope and market size – e.g., virtualization for x86 CPUs. 2) Medium-sized, often enabled by a new, now-feasible hardware form factor on which software can run and generate value – e.g., smartphone apps. 3) A massive architectural paradigm shift – e.g., cloud computing.

Regardless, of size, as NewSoft unfolds, the smart money says that at least one of the following categories of software opportunities are not far behind:

- Automated deployment of NewSoft at scale, because if NewSoft is successful, some aspect of it will initially create more, not less, work.

- Integrating legacy apps with NewSoft, which frequently starts to take off when NewSoft customers move from early adopters to the mainstream, where customers are more averse to getting rid of the previous legacy "OldSoft" even as they want to begin deploying NewSoft.

- Securing NewSoft – usually a safe, if short-lived NewSoft-tangent bet because developers rarely build software to be secure from the get-go but rather to be promiscuous and maximally compatible to promote rapid adoption (fintech represents an entire category of exceptions to this).

- Service Level Management (SLM) for NewSoft to enforce and monitor Service Level Agreements (SLAs) when NewSoft is used in more mission-critical situations to reduce risk for mainstream adopters and satisfy their objections with monetary guarantees.

- Consolidated/dedicated management of NewSoft enabling some strawman administrator to get rid of all the ad-hoc tools used to handle NewSoft management in pinch. At every software conference, there is always a vendor imploring you to "get rid of the spreadsheets you’ve been using, consolidate your tools, and adopt their single pane of glass solution"… so many "single panes of glass"…

Whether it is a truly novel innovation made possible by the latest jumps in leading edge Tr/mm^3/$ or by declines in trailing edge $/Tr, or whether it is one of these long-cycle "rhyming" innovations, if, as an investor, one does not know what might be coming, one might not believe it when it arrives. In turn, chronically passing up these opportunities, which frequently result in "multi-X" returns over 10-year, 5-year, and even 3-year investment horizons, leads to missing out on considerable alpha generation. In trying to figure out what is coming in future states of Tr/mm^3/$ and Tr/$, we thought we would share the following graphic diagrams and associated visualizations that have been helpful to us over the years.

As discussed in Rule #1, these are not meant to be formulaic but intended to stimulate more rigorous thinking about innovation, and they are always works in progress. As such, we welcome constructive feedback on how they might be improved or extended to encompass a wider variety of real-world situations.

CLAYTON CHRISTIANSEN'S DISRUPTIVE INNOVATION IN THE CONTEXT OF MOORE'S LAW (PART 1)

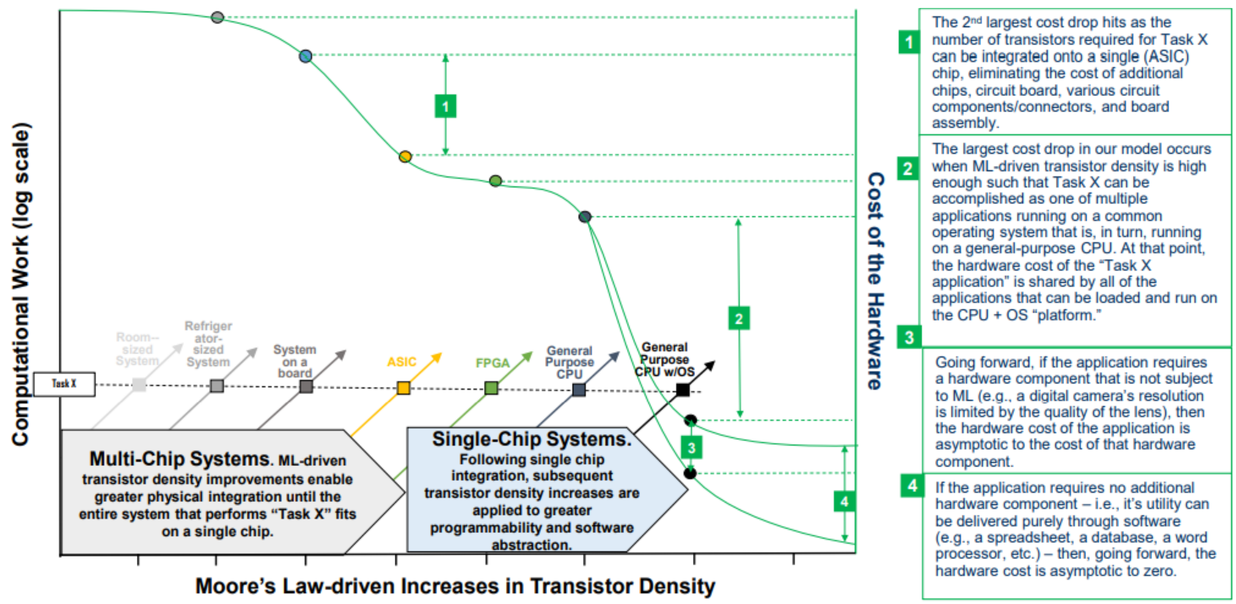


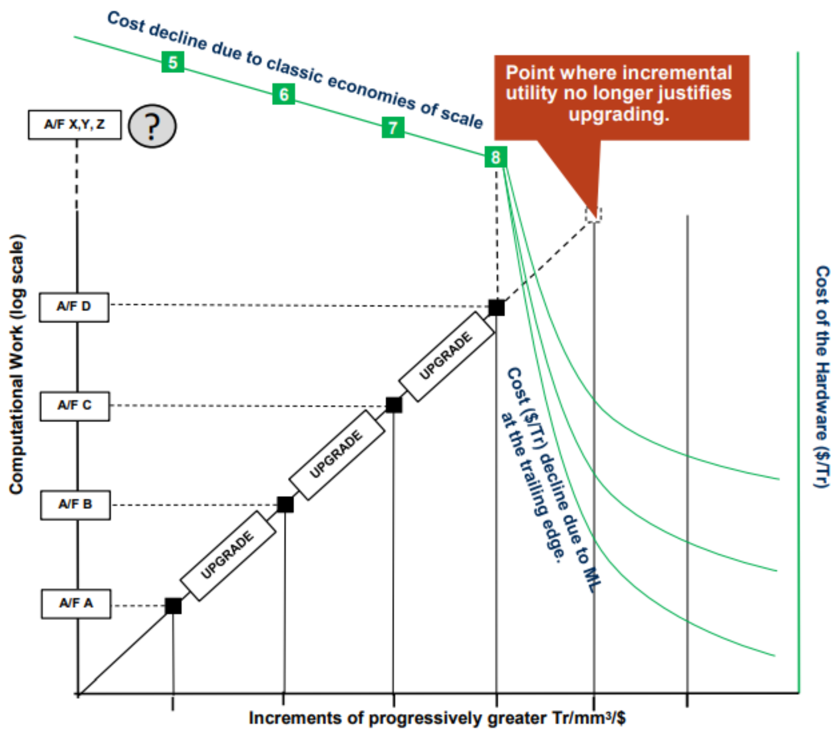
Diagram Explanation

As transistor density increases, so does computational work that can be applied to "Task X." Task X could represent a wide range of applications, such as a firewall 1Gbps or a conditional 3-field sort of a database with 100,000 records, etc. However, in practical terms for investors, it can be thought of as an aggregate of tasks culminating in the ability to have a touch screen-driven smartphone in your pocket with enough processor headroom to support an eco-system of third-party apps. Initially, Task X can only be accomplished using multiple processors on multiple circuit boards. In the "increasing integration" phase, Moore's Law (the rate of transistor doubling being expressed as the slope of the curves for each of the form factors) eventually enables execution of Task X by a single custom chip or "ASIC" – though some "tasks" do not lend themselves well to a lack of programmability, others, like Ethernet switches, do. So, the answer to the question "does programmability add value here?" is often key.

In the "increasing programmability" phase, Task X is executed by processors of increasing programmability. In the final phase, Task X becomes simply one of many applications running on an open operating system (OS), in turn running on a fully programmable processor (i.e., general purpose CPU but fully programable GPUs and DSPs also qualify). Depending on the scope and applicability of Task X, each increment of greater transistor density opens the door for innovation, but the next one opens up other doors for disruption by competitors. However, once it is running as an app on a fully programmable processor, the odds of disruption fall as the app can then ride through time on the increasing transistor density – aka computing capacity – of the processor as new versions come out, thus protecting itself, to some degree, from Moore's Law-driven disruption.

At that point it moves to the "sustaining innovation model" diagram below for platforms where each Moore's Law-driven increase in transistor density adds value to what is now a platform by enabling more apps, usually third party, for maximum total "gravity of value," or functions (i.e., A/F) until that platform has "run its useful course." At that point, an entirely new platform becomes feasible and takes center stage. However, once established, even old platforms can continue generating value long after its ecosystem has decayed, most of the constituents have left, and it has lost most of its gravity. Sometimes though, there is simply a lag before the next threshold of Tr/mm³/$ or $/Tr is crossed, and we see an explosion of new apps and functionality to revive the platform and catalyze another round of upgrades.

CLAYTON CHRISTIANSEN'S DISRUPTIVE INNOVATION IN THE CONTEXT OF MOORE'S LAW (PART 2)



Identifying true innovation and gauging its value before it begins to fully play out is the most difficult, hard-to-pin-down task when making investment decisions regarding software-based businesses. And that's a good thing… at least for this analyst. As soon as it becomes formulaic, software will certainly substitute for labor in the means of production for equity research. In the meantime, if there is a rule inside these "not rules" of investing in software-based businesses, it is this simply obvious mantra: know software. Know as much about software as you can stand to cram into your mind – software's history, its personalities, its breakthroughs, the history of the software industry and software-based businesses, as well as the history of industry and business in relation to software.

RULE #5. THE BLACK HOLE OF COMMODITIZATION

Previously, in Rule #4, we discussed our idea of what innovation means, about understanding software's inseparable role in innovation today, identifying innovation early, analyzing the company that produced it, and characterizing its addressable market, profitability, sustainability, and extendibility in the context of geometric annual increases in the innovation's fuel: computing capacity per unit cost. As one can see already, many of these "rules" represent different takes or perspectives on attaining success in these endeavors.

Over the last 40 years, we have seen an increasing number of examples of software-based businesses generating benchmark-shattering "many x" returns over the long-term – two to eight years being the sweet spot, in our view – for equity investors who can recognize innovation early. However, that is becoming an increasingly difficult task in an industry where all companies proclaim themselves innovative. In this big, ambitious set of tasks, we find it helpful to model and visualize innovation's opposite – commoditization – and we like to think of it operating like gravity. However, while the Black Hole of Commoditization (BHoC) primarily pulls in hardware, the impact of gravity in our astrophysical model differs depending on the mix of and degree of integration between hardware and software in a company's products. We model six types of companies based on degree of hardware and software integration, size of addressable market, and status as a "platform."

Wait! All this stuff about long-term horizons, patience, innovation, and geometric growth sounds great, but MSFT shares were dead money for almost 15 years. Then, from 2015 to 2021, the stock generated a 6x return and produced over a trillion dollars in aggregate value. What was the difference between 2001-2015 and 2015-2021, and could the later move have been anticipated?

One "surface" look at MSFT in 2015 would have one believe that its best days were over. In 2016, MSFT revenue actually declined Y/Y. It certainly seemed like "ole softie's" trajectory was fated, yet another example of a tech sector incumbent missing all that is new, relegated to milking existing franchises, making retrograde acquisitions to enhance that milking, and levering up to buy back shares to sate existing shareholders and inflate EPS, engineering just enough earnings growth to qualify as "GARP" by generalist institutional investors even though revenue flatlined since 2013, even with several acquisitions... but enough about Oracle. We will come back to MSFT in a bit.

What Happens to Hardware in the Black Hole of Commoditization?

Before we get to the six company types, we want to explain, in general, the aspects of gravity that remind us of the process of commoditization and introduce a particularly influential framework for understanding the adoption of technology that has been around since the 1990s.

First, no matter how initially innovative and successful a pure hardware product – device or entire system – may be, profits become crushed by abnormally heavy price-based competition just as unit volumes soar. Theoretically at least, even gross margin eventually disappears in a profit singularity inside this "BHoC". In our view, this occurs not in spite of but because of soaring unit volumes. It often seems that when a hardware product "crosses the chasm" of the adoption curve, it catalyzes its own commoditization.

In *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers*, one of the early classics about how the tech industry operates, author Geoffrey Moore presents his influential insights linking the tactical sales process and the trajectory of a new tech product, whether hardware or software, to a high-level framework. The "chasm" to which Moore refers in his otherwise normally distributed, bell-shaped adoption curve exists between, on one side, early adopters, comprised of "technology enthusiasts" and "visionaries" representing 15% to 20% of a given market, and, on the other side, the much larger mainstream market, with "pragmatist," "conservative," and "skeptic" customers to whom it is hard to sell new things. When a product is validated by a small number of influential, "pragmatist" customers in the early mainstream, it is a very good sign that it has "crossed the chasm" and is now set to quickly take the rest of the mainstream market. This is when unit volumes really take off.

Second, Big Tech's FoMo (Fear of Missing Out) also often kicks in when a new piece of hardware hits the steep part of the adoption curve, prompting a few of the "Big Tech" players to place a few of their many chips down and build something. Often, Big Tech and other large competitors just want to gain a foothold or at least a presence in an area that could end up being completely game changing and strategic. For the large, software-based businesses we call "Big Tech," subsidizing a hardware product – be it a video game console, a TV, a thermostat, or an army of smartphone clone makers – is often a smart way to play defense while a trend unfolds. Once adoption saturates and the specific segment related to the hardware evolves, a Big Tech player's involvement on the hardware side may give them an advantage in finding the more sustainably profitable opportunities in that segment.

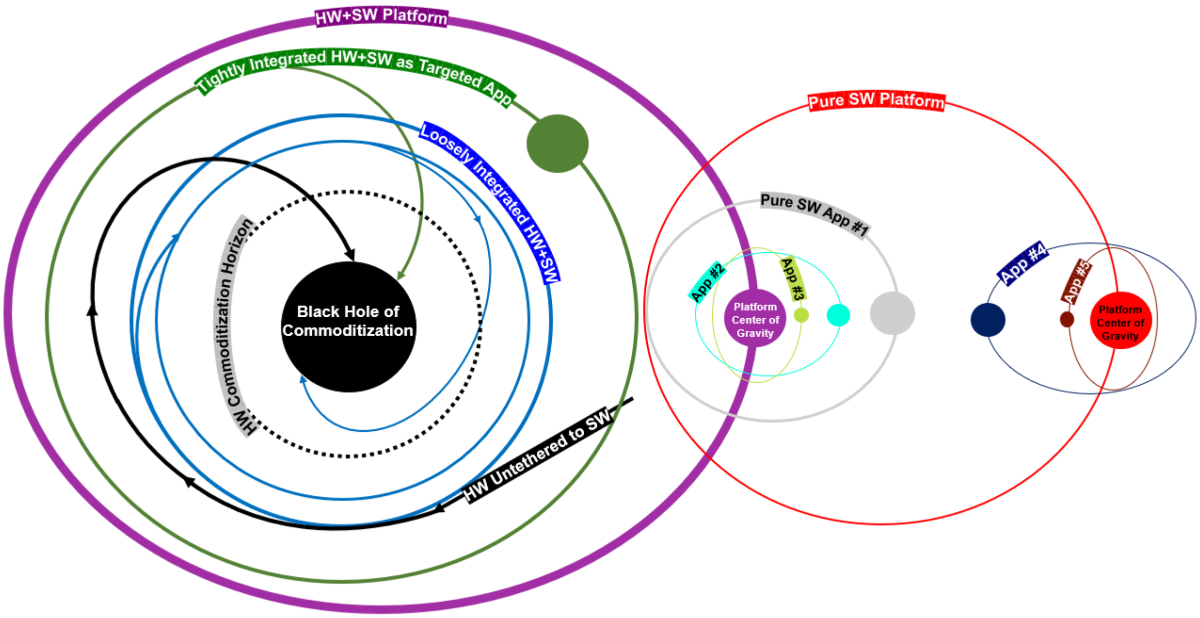
Third, whether it's hardware or software, if a product is on the right side of Moore's Law (see Rule #4), crossing the chasm into the mainstream often coincides with big, important upgrades relatively early in the product's lifecycle for early adopters even before the product crosses the chasm into the mainstream. This represents another accelerant to unit growth and, ultimately, to the BHoC's gravitational pull. Untethered hardware can sometimes circle the BHoC a few times in low margin orbits, but it requires flawless execution by management of a clever brand-focused strategy. Yet, no matter how innovative the design, if a hardware product is not tethered to software, brand cachet cannot resist descent into the BHoC for more than two extra Moore's Law cycles or so.

Price-based competition for hardware can get so intense and the gravitational pull of commoditization so strong that, as greater scales of volume production are attained, any operating leverage generated is eaten up immediately by all these new, increasingly loss-tolerant entrants, attracted by the exponentially increasing market that is already much bigger than anyone expected. Lastly, hardware segments often see the entrance of component suppliers with a cost advantage through vertical integration – e.g., Samsung in smartphones, where it can sell itself ~90% of the semiconductor content (mobile processors, memory, and solid-state storage), not to mention the OLED screens.

There are other aspects of gravity that remind us of the process of commoditization. For example, as is the case in the real world too, the role of the observer – aka customer – is pivotal. Once a product is declared by a few vocal customers to be a commodity, it becomes a commodity, and we describe it as having crossed the Commoditization Horizon, with all of the implications of that as a "point of no return" being fully intended. In the retail market, few consumers tolerate both a price premium and a lack of differentiation for products that cost hundreds to thousands of dollars. And few CIOs or CFOs, whether at a large enterprise or small to medium-sized business (SMB), want to be accused of wasting money by overpaying for a commodity.

Below is our first diagram showing the six types of companies with different mixes of hardware and software, each with its own orbit. Following that, we have our descriptions of each type, why they traverse their particular orbits, and some historical examples.

THE BLACK HOLE OF COMMODITIZATION AND THE ORBITAL PATTERNS OF VARIOUS SOFTWARE-BASED BUSINESSES (PART 1)



Source: CFRA.

1) Hardware untethered to software (clones!), reliant on, and given value by 3rd-party software. Since we have already described the BHoC phenomenon for untethered hardware, here we present two well-known historical examples. Our first occurred in the early 1990s, just after Intel produced a standardized PC motherboard, basically giving a template to anyone with a soldering iron and a garage to become a PC maker. At about the same time, Intel moved to a more advanced manufacturing process for its 486 CPUs enabling 1-micron-sized transistors vs. the 1.5-micron process it used for its first 386 CPUs in 1985. Thus, the 486DX2 CPU featured a transistor density of 6,822 Tr/mm3 vs. 2,644 Tr/mm3 for the 386 (source, Intel). This enabled Intel to fit both a math coprocessor and L1 memory cache on the 486DX2. This, in turn, allowed a more stable, faster version of Windows to supersede the command-line-driven DOS operating system, which was more processor and memory efficient but much less intuitive and user-friendly.

Soon, like bacteria in a petri dish left out overnight, hundreds of "Intel Inside" PC makers – assemblers might be a better word – entered the PC market, each undercutting the next on price and all trying to "make it up on volume." The rise of "Intel Inside" enabled the WinTel duopoly and its integrated hardware and software platform to hollow out and subvert the "IBM Compatible" grip on the market and capture the lion's share of value from the platform. This led to massive market share and operating losses for IBM and other incumbent IBM "clone" PC makers worldwide, including Group Bull, NEC, DEC, AST, and Fujitsu as they plummeted rapidly into the BHoC. At the same time, commoditized but still computationally powerful PCs also produced an explosion of Windows applications, WinTel compatible peripherals, and an entire eco-system with a counterbalancing anti-gravitational pull of profitable software.

Interestingly, one lyric that rhymes across the different states of Tr/mm^3/$ in the history of software is the furious denial by hardware-centered companies that they have indeed crossed the Commoditization Horizon and are destined to fall into the BHoC, despite the clarity of that reality from the perspective of any outside observer. Perhaps, because customer demand and unit volume are usually also increasing simultaneously, the acceleration felt while falling into the BHoC cannot be differentiated from the acceleration felt when rocketing up into a higher, more profitable orbit. Or maybe this is just a case of an equity analyst pushing his metaphor too far...

2) Hardware and software bundles integrated too loosely to generate incremental value together. In these cases, the software must compete on its own merits eventually, unaided by a performance differential or features that come from tighter hardware and software integration. The hardware product, as novel a use of the most recently delivered computing capacity surplus as it might be, will end up commoditized as soon as customers and competitors realize the looseness of the coupling and no longer place value on the combination. This typically happens after a delivered surplus in computing capacity at a particular form factor makes greater hardware standardization feasible and when the value of that standardization exceeds the value of maintaining proprietary hardware design and/or increasing proprietary advantage through greater hardware and software integration.

Novell Netware and its bundled Ethernet Network Interface Cards (NICs) in the late 80s and early 90s come to mind here. The bundle served a systems integration function early in the adoption of Local Area Networks (LANs) to make it is easier for distributors and customers. Once a little extra processing capacity became available, Intel's motherboard template helped standardize LAN interfaces, connectors, and cables making NIC installation easier and driving down NIC prices. Novell recognized what was happening early and abandoned the NIC market, focused on its Netware software, and promoted compatibility for any certified Ethernet NIC vendors.

When that decision was made, Novell was primarily battling IBM in the LAN market, with all of IBM's proprietary networking standards, connectors, and even cables. However, it was MSFT, specifically Windows NT, that took down the early dominance of Netware in LAN and server operating systems. MSFT developed and added code to NT to assume network operating system duties along with being a server operating system. Of course, the cheaper, much more powerful "Intel Inside" servers accelerated NT's takeover, taking Novell's management by surprise. More recent examples, in our view, include GoPro and Nutanix, the latter making the move to a pure software model just in time (perhaps a little late but better than never), and the former sticking with selling hardware products and now struggling against the pull of BHoC, likely to no avail, in our view, as it has already crossed the Commoditization Horizon.

3) Tightly integrated hardware and software focused on a specific, high value, often computationally difficult application. These hardware and software combos can generate value for years, seemingly well above the fray in a high orbit. Then, we hit a state of Tr/mm^3/$ where the computational work of that combination of tightly integrated software and hardware can be handled as an app, one among many, running on a general-purpose system. At that point, the plunge into the BHoC can occur with shocking speed, as the fast falls of Research in Motion (RIM), Nortel, Lucent, Avaya, and many other telecom equipment and enterprise PBX systems can attest.

RIM's story is especially illuminating here, in our view. RIM's product, the BlackBerry smartphone, evolved from RIM's late-90s devices featuring truly innovative and optimal use of the Tr/mm^3/$ available at the time, clever hardware designs, a deeply integrated software operating system, and even an associated messaging relay service that RIM operated from its own data center. RIM's tightly coupled hardware and software product was first available in the form factor of a cellular pager. Within the first decade of its existence, RIM's BlackBerry devices were dedicated to a singular purpose: sending and receiving email while mobile, with messages traversing the narrowband, high latency, and unreliable connections of 1G and 2G cellular networks and dial-up internet. Delivering mobile email was no mean feat – until it was.

In about 2009, there was enough surplus Tr/mm^3/$ to enable a general-purpose platform, the Apple iPhone, to support a growing software ecosystem, including apps to provide secure, reliable email and a host of other third-party apps running on its operating system, iOS. Suffice to say, creating and distributing third-party apps was never easy on RIM's operating system, software rooted in an age when computing capacity was at a premium and getting a cellular pager to receive email required much more economical use of that capacity. In 2010, we also had much faster 3G cellular networks, soon to be upgraded to be 4G, and increasingly ubiquitous broadband Internet access via WiFi. The game "Brickbreaker," which perfectly leveraged Blackberry's physical trackball and became very popular with Blackberry users, is a notable exception.

Like in the allegory expounded upon in Rule #3 with the Emperor of India who does not feel the threat of geometric growth until it is too late, RIM's reaction to the iPhone seemed cloaked in a "light fog of denial." So frequently, this is the dismissive approach that takes over as a defensive default among the leadership of even the most competent hardware companies hurtling toward commoditization or experiencing one of the other results of disruptive innovation. Some examples...

> *I've been frankly confused by this fascination that everybody has with Netflix...Netflix doesn't really have or do anything that we can't or don't already do ourselves.*
> **– Jim Keyes, CEO of Blockbuster Video in 2008**

> *Smartphone makers who use Android are like boys from Finland who "pee in their pants" to keep warm in winter. It may feel good at first, but then things get worse.*
> **– Anssi Vanjoki, former EVP at Nokia in 2010**

> *There's no way that company exists in a year.* [That company was none other than Salesforce.com.]
> **– Tom Siebel, founder and former CEO of Siebel CRM Systems in 2001**

> *The computer industry is the only industry that is more fashion-driven than women's fashion. Maybe I'm an idiot, but I have no idea what anyone is talking about. What is it? It's complete gibberish. It's insane. When is this idiocy going to stop?* [Referring to cloud computing in 2009]
> **– Larry Ellison, founder and former CEO of Oracle**

As late as 2010, RIM executives continued extolling the advantages of their "more secure" email with sub-second delivery latency compared to the 2.5 sec average delivery for one of several email packages made available on the iPhone. If there is a more specific lesson here, we suppose it is this: with any new successful platform, security is an afterthought; access promiscuity and broad compatibility win every time, at least initially. Then, security solutions emerge to fix the problems that begin multiplying when "access promiscuity" begins to scale. From the Green Caterpillar UNIX virus to Bitcoin, it's one of the oldest story formats in the screenwriter's guild for the software industry.

What RIM did not do, which Apple did, was create a platform for third-party app development and grow a massive developer community. Other examples here include the telecom equipment providers such as Nortel and Lucent with entire product lines of Class 5 and Class 4 switches that made up the circuit-switches public telephone network, featuring proprietary combinations of very tightly coupled software and hardware with very little ability to integrate with potentially complimentary third-party solutions. Even after those solutions were acquired, as dozens were in the late 90s, Lucent and Nortel struggled internally to integrate them with their existing product lines.

4) Tightly integrated hardware and software as a general-purpose platform and supporting a range of applications. This is the Holy Grail of hardware aspirations, with hardware design core competencies interlaced with dependent software functionality – not to mention an eco-system of interdependent third parties and certified professionals that establishes an especially difficult competitive moat to cross. Successful Type 4 hardware franchises have more ways of generating sustainable competitive advantages and generate higher profits for longer than software-only franchises. As such, attaining this type of integrated hardware-and-software status is very difficult, expensive, time-consuming, and rarely successful.

However, once established, the aggregated gravitational pull of the constituents with a vested interest in the success of the platform can easily counterbalance the pull of commoditization. These constituents can include third-party software providers, peripheral makers, consulting firms, and the population of professionals that have built careers around being experts on the platform or communities of experts for each major app on the platform. Rich platforms formed by this tight coupling of hardware and software can generate considerable anti-gravitational leverage resulting in long-duration competitive advantages, pricing power, and sustained profitability.

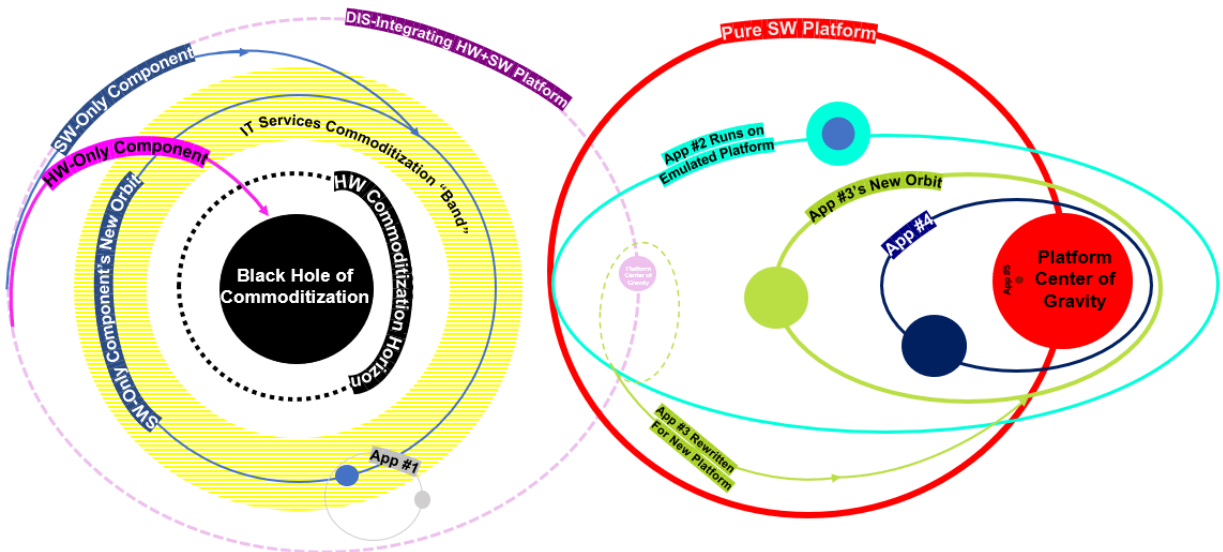
Apple has built and maintains two of these Type 4 platform franchises, the Mac with OSX and the iPhone with iOS. Microsoft has built one by itself (Xbox) and has shared the other with Intel since the 80s (the "Intel Inside" PC with Windows). Cisco also qualifies here with its IOS (Internet Operating System) routing and switching software tightly integrated with its routers and switches. Although in Cisco's case, "general-purpose" is still confined within the realm of networking. Obviously, while very successful within networking, Cisco was never able to parlay its dominance into larger, more truly general-purpose computing markets.

5) Software applications. Here we are considering only traditional, license-based software applications deployed on-premises and running on customer-owned/controlled devices. The fate of apps and their orbits depends on more than just the success of that app, the value it brings vs. competitors, the size of the addressable market, etc. While those elements are crucial, the success of a software application also depends on management decisions that have both tactical and strategic considerations regarding platforms, programming languages, development frameworks, architectures, modes of deployment, and so on.

Further, the success of an app depends on the success of the chosen platforms, languages, architectures, etc. In this way, large numbers of companies with single apps or suites of related apps – even those that are direct competitors – form broad, powerful alliances to promote interoperability and the success of the chosen platform, language, etc. Since they all share in its success, they often have more incentive to cooperate, at least at some level, than to compete, at least directly. The vast number and variety of dot.orgs, alliances, and partnerships among software companies testifies to this.

As one can see from the second BHoC diagram below, software companies have ways to defy gravity and protect themselves from commoditization in ways that hardware companies cannot. First, commoditizing hardware is to the software industry what plummeting gasoline prices are to auto manufacturers. Other variables held equal, the geometric growth in computing capacity per unit cost improves the customer's return on investment (ROI), thus encouraging greater investment and consumption of software by the customer. Yet, such falling prices and growing capacity also open up entirely new applications and new computing form factors (e.g., a smart watch) for a software app to run on. In most situations, apps benefit from and "gain orbit" when hardware falls into the BHoC.

THE BLACK HOLE OF COMMODITIZATION AND THE ORBITAL PATTERNS OF VARIOUS SOFTWARE-BASED BUSINESSES (PART 2)



Source: CFRA.

Also, fueled by Moore's Law, software is driven to ever greater degrees of abstraction, allowing it to adapt, jump, and migrate to new platforms, paradigms, and even eras – if management is on top of it. Sometimes, a company can simply rewrite an app for a new ascendent platform (e.g., App #3 in Figure 2). Although that approach rarely ends up being "simple," it can enable a better, cleaner break from a previous platform and make the most sense long term. Other times, apps can be ported to a new platform if the company or a third party writes software that emulates the old platform on the new one (e.g., App #3 in Figure 2). While wasteful of computing capacity, it allows multiple apps to be ported to the new platform, all running on the same emulation software. As one might have guessed, because of Moore's Law, that wasted computing capacity becomes a moot point at some point.

However, even if it fails to do so, an app can remain on an old platform, sometimes even just the software component of that platform (e.g., App #1 in both Figures) for years, even decades, depending on the mission critical nature of the app and the risk tolerance of the customer base, without falling into the BHoC. Due to the conservative "if it ain't broke, don't fit it" axiom, invariably joined by the power of habit and human inertia, companies can profitably maintain apps on old platforms running on the latest commoditized hardware for a very long time. Instead of plummeting to zero gross margin, an apps business will often evolve into an IT services business for an installed base of code, often too embedded to easily replace. The bad news is that the operating margin of a software business that has fallen into a lower band becomes asymptotic to that of an IT services business, but it is usually a gentle fall from operating margins in 30s typical of a software company at peak profitability. The good news is that this type of business can be profitable all the way down to the last customer with the app and the platform still installed.

Sometimes the stickiest part of a software app is not about platforms or its links to other software but with its links to people, the app's users. In many open-ended and surprising ways, the linking and reinforcement that occurs between human cognition and software is as strong for most people as that same link is for a master carpenter and their physical tools or a U.S. Marine and that Marine's rifle. Here, we think it is very important to monitor the buzz in the industry about the value of certain certifications and what programming language might be showing rapidly rising interest among communities of developers.

6. Software platforms running on either a hardware and software platform or on another pure software platform. Similar to a Type 4 hardware/software platform, pure software platforms are harder to establish and gain traction with than apps, but, once a critical mass is achieved, the entire mass of their planetary system can counterbalance the gravitational pull of the BHoC, even if the pull-on software can only lead to lower operating margin of an eventual IT services business. In the past, most platforms were operating systems. Oracle's database software, originally built to run on IBM's mainframe operating system, beat out several competing database platforms during the client server era (e.g., Informix, Sybase, Gupta) to take leading market share in databases, with only IBM's DB2 and MSFT's SQL holding significant share by 2005. Instead of falling into the BHOC, even large planetary systems can decay over time if they no longer represent the best way to take advantage of additional Tr/mm^3/$ to add value, hence fading and losing apps to ascendent platforms.

Speaking of software platforms... that brings us back to MSFT in 2015, because, if you looked just a little bit closer at how that company had changed when the board promoted Satya Nadella to CEO in 2014, one may have seen the changes that were coming and gotten in on those shares before its big move from 2016 through 2022. Nadella was the former EVP of Enterprise and Cloud Computing, the man who was given substantial credit for building the Azure cloud services business, the first real and still largest competitive threat to Amazon's AWS. With Nadella's promotion and given his articulated vision, MSFT was deemphasizing mobile in order to boost its focus on cloud services. In doing so, MSFT was able to ride the cloud migration wave over the last five years. Today, most of MSFT's businesses are clearly on the right side of Moore's Law, positioned for a better era-hopping parlay of its existing strengths compared to trying to turn Windows into a successful mobile operating system.

Azure was differentiated from the get-go, forward-looking, and likely to continue improving and scaling more than proportionately in future states of Tr/mm^3/$ and Tr/$. It was also launched while Amazon AWS was still "pre-chasm," enabling MSFT to establish Azure as the number two cloud infrastructure and platform offering. Further, MSFT's "cross the chasm" strategy was well conceived and well-executed. Azure was designed to aid the migration of apps built with .Net tools, engaging a massive network of MSFT-affiliated developers and custom code that MSFT has under-monetized for decades. We note that the Microsoft Visual Studio app development suite was the precursor to the .Net framework and dates back to the 1980s. By targeting that large number of .Net developers and the installed base of .Net apps, Azure was able to quickly gain momentum to cement that position.

Though an early cloud version of Office was made available to corporate customers as early as 2011, Office 365's initial "comprehensive" launch did not happen until 2017, along with a sales and marketing campaign to get customers to switch to Office 365 from their legacy Office Suite. This conversion process meant that MSFT would face a tell-tale, cloud-shift headwind to growth until cloud-based subscriptions exceeded revenue from license sales and maintenance contracts on the installed base of Office Suite. After that point, which MSFT likely hit with Office 365 in 2019, the headwind became a tailwind as MSFT is now receiving much more revenue per Office customer; it is just that this higher revenue was spread out over time into the future.

Also, it was clear that enterprise customers and other cloud-based app providers had a strong incentive to give business to a legitimate number two provider, in this case MSFT.

Since Amazon's launch of AWS, two of the remaining three hyperscale data center operators, MSFT and Google, have used that scale to launch cloud services. Though the smallest of the three, Google Cloud is differentiated by its focus on deep learning via its TensorFlow development framework and the associated custom processor used in some of Google's servers. In fact, given the size and growth of the opportunity in cloud infrastructure and platform services, and the fact that the sheer scale of computing capacity enables only a handful of competitors, we wonder, what is taking MetaFace so long? But we digress.

Though few could have envisioned the magnitude of MSFT's upside, certainly, even at the time, it was easy to see that a cloud-focused revamp of MSFT was a higher probability with a clearly higher monetization potential than any rationally envisioned payoff on the mobile side, especially any scenario involving Windows Mobile. Nothing is set in stone, and there is no such thing as fate in the software industry, as much as it might look that way retrospectively. Yet, by 2015, there were signs that MSFT could very well be turning a corner with growth set to reaccelerate. MSFT's flatline stemmed from a corporate culture infested with rear-view-mirrorism and an instinct to simply copy Apple's latest innovation. Yet, since Windows, MSFT did not even do that especially well. MSFT kept zigging when new industry leaders, especially Apple, were zagging, seemingly always a step behind.

For example, MSFT used the surplus transistor density per dollar of the late 90s to deliver, in 2006, the Zune, a hardware device with a music app five years too late and just before Apple was to change the portable music device game again – by turning it into an app running on a general-purpose hardware and software platform, i.e., the iPhone and iOS. It is hard to find a better example of being on the Wrong Side of Moore's Law than the Zune.

Yet, MSFT kept doubling down to maintain some foothold in the mobile space with launch of the "Kin" smartphone line in 2010 and the acquisition of Nokia's handset business. Yep, entering the handset business in order to get into the smartphone business is a strategy that is also clearly on the Wrong Side of Moore's Law, in our view. Then, MSFT got "out-Microsofted" by Google in the mobile space, what Bill Gates called "Microsoft's greatest mistake." This mistake gave rise to what is today an operating system duopoly in mobile devices that does not include MSFT, despite it holding a monopoly or duopoly in operating systems for PCs, enterprise servers, and video game consoles. In order to effectively mix human and software cognition to produce innovation, companies still need focus, even when they are diversifying their bets. MSFT's comeback and the 6x return the stock generated from 2015 through 2021 were due to the twin successes of cloud service offerings, Azure and the "365 versions" of its Office Suite and related apps.

Not all aspects of pre-2014 MSFT were bad though; some developments just took a long time to bear fruit. Obviously, in addition to Azure, which was started in 2010, MSFT's ongoing commitment to search engine Bing, which is likely profitable even at only ~6% of the U.S. search advertising market share, probably contributed considerably to MSFT being able to justify building and operating hyperscale data centers, as did the addition of LinkedIn a few years later. Xbox Live also contributed to attaining hyperscale. Today, MSFT qualifies as one of the four "hyperscale" cloud operators in the U.S., the other three being Alphabet, Amazon, and Meta. Achieving hyperscale before Azure really took off was not a prerequisite for MSFT's cloud-based transformation, but it was very helpful, in our view, as a way for the company to economically build out Azure's now massive catalog of services and launch both Office 365 and Dynamics 365.

For us, the first sure sign that MSFT was moving to the right side of Moore's Law was the promotion of Satya Nadella, former EVP of Cloud and Enterprise, the man most singularly responsible for the early success of Azure, to CEO. Soon after, Nadella abandoned MSFT's mobile ambitions and began the company-wide focus on and transition to cloud. We point out that there were several years between these strong "pro-cloud" moves in 2014 and 2015 and their payoff in the fundamentals and then even a little longer in the share price. In MSFT's case, there was plenty of time for equity investors to spot the pattern, validate it, and make the investment... that is... if one were paying attention to software and spending less time trying to figure out whether to call MSFT "value" or "growth."

So, where do cloud app providers fit? As we began to visualize a cloud company, we realized that most of the successful ones have built large planetary systems that are implied in and affected by many other systems. Visualizations based on gravity work best when modelling things that operate in either a high degree of orbital stability (e.g., the high orbit of a traditional, legacy client-server application) or one that could descend quickly into the BHoC like GoPro. With cloud and the MicroServices Architecture (MSA), now standard for new cloud entrants, there are so many more abstractions and so many more standardized ways for different cloud apps to talk to each other, exchange data with each other, or access certain software functionality on the fly. This later capability has been made especially easy at the present state of affordable computing capacity, which has enabled much more explicitly federated functionality with capacity to spare. A good example here is Twilio (TWLO). TWLO's cloud-based app handles real-time comms and messaging functionality for any other app, as a subcontracted service, as easy as drag and drop for app developers.

While all of this certainly generates substantial "gravity," there is clearly a much more dynamic and multivariate aspect to visualizing cloud vendors and trends. The equilibria among competitors can be stronger and longer lasting than in previous eras while, at the same time, more dynamic and evolving. It is almost as if there is standard competition among companies for customers for specific software apps, which remains quite active, no doubt. Yet, at the same time, a simultaneous and related game is intensifying among software companies to see who can combine their software with the right business model and tactics and right partners to generate and capture the most value within the eco-system. Today, that game is emerging as the primary Darwinian imperative among software companies.

So, while mapping cloud-based software companies, as well as software-based businesses not in the software industry, onto this astrophysical metaphor, there were many more orbits to be drawn and levels and types of connectivity and influence with alternating states of attraction and repulsion, competition, and cooperation. We needed a new metaphor to capture the complexity of the cloud era, one with more of a biological inspiration. But that's for a later rule.

RULE #6. WHAT DIGITAL TRANSFORMATION MEANS

Digital Transformation is Software Substituting for Labor (Usually) and Capital (Sometimes) in the means of production. The present trend called “Digital Transformation” – the subject of countless books and articles about Information Technology (IT) in the last few years and one of the hottest subjects at every MBA program these days – can best be seen in the context of the evolution of enterprise computing and software since the beginning of IT itself during mainframe era of the ‘60s and ‘70s. If you want to know the nature of a thing, it is critical to know its history.

In the previous century, software made its greatest impact in automating and dramatically improving the ability of enterprises to manage the means of production. Enterprise software enabled traditional, industrial era, functionally “mechanical” companies to attain much greater organizational scale than what was possible with thousands of mid-level managers and accountants armed with clipboards, pencils, and mechanical adding machines – the mid-level bureaucracy and organizational complexity posed limits on scale. Beginning in the ‘60s for mainstream companies and other large enterprises, the increasing deployment of software running on room-sized mainframe computers that possessed far less computational capacity than todays’ smartwatches. Nevertheless, software running on even this small amount of computational capacity went a long way to aid and automate the human cognitive load as it was applied to the management of a growing variety of large enterprise organizations, enabling these organizations to wring ever greater efficiencies from their operations, and breakthrough those limits to scale.

What we are seeing today with digital transformation is profoundly different; even knowledgeable people have a hard time describing exactly what that difference is, especially in a generalized sense. They struggle with the question: “What makes digital transformation different from enterprises simply buying and deploying more and more software as they have for decades?” The following represents our answer to that question. However, we want to stress that the following is also only one answer from this analyst in the context of investing in software-based businesses and is meant to be the beginning of the discussion, not the end.

As it is with “the Dao” (loosely translated as “the way”) and as described in the Dao De Ching, the digital transformation that can be named is not the real digital transformation. There are as many definitions of digital transformation as there are PowerPoint presentations about digital transformation. Nevertheless, we think our definition and historical take on digital transformation might be helpful from an investor perspective, but we absolutely acknowledge many other perspectives on the subject.

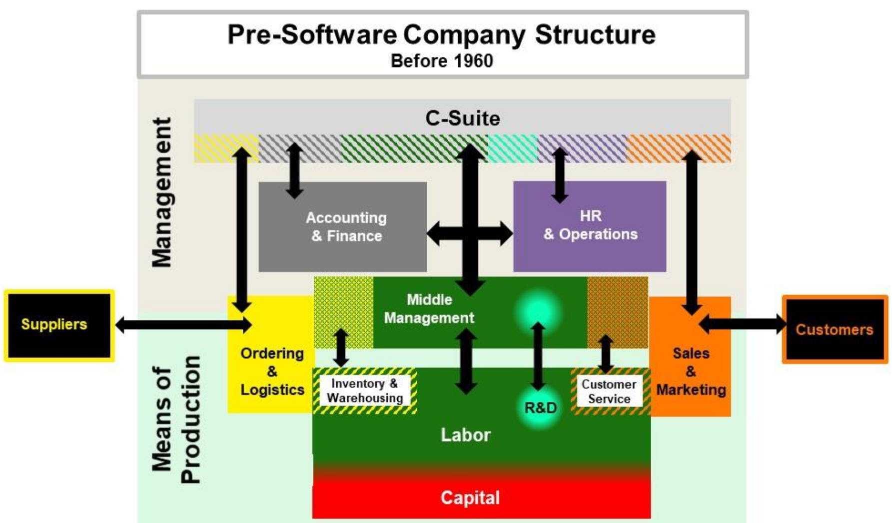
In our view, in digital transformation, the primary purpose of software is not to automate or aid in the management of the means of production, though it often helps in that regard. As we see it, in digital transformation, software is the means of production. Software investments are technically a form of capital investment. However, even that is a poor categorization, as software investments notoriously entail complex accounting governed by a wide range of fuzzy rules and ambiguous standards. Software is clearly a very different animal compared to real estate, warehouses, a fleet of trucks, or a gas-fired generator powering a factory.

In the most generalized sense and borrowing from the field of microeconomics and the behavior of the firm, digital transformation projects entail the substitution of software for some combination of labor and capital in the means of production.

The Pre-Transistor World of “Paleosoftware”

In the diagram below, we depict the “initial state” of the evolution of the enterprise organization when software would be barely recognizable as such today – the product of human cognition and processes instantiated in biological memories, individual and collective, supplemented by books, files, pamphlets, memos, pictures, maps, and a variety of other analog methods for data capture and retrieval.

However, for the more numerically oriented cognitive tasks, there was a growing role for digital calculating machines using a combination of mechanical and electric relays, though this was still well before the emergence of IT departments dedicated to the construction and operation of software.



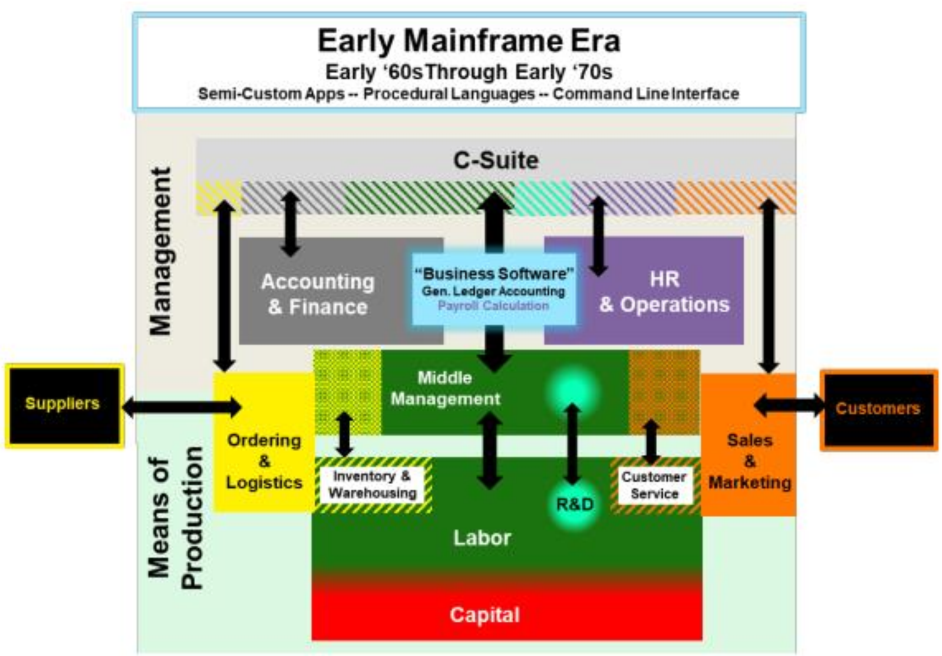
Source: CFRA.

Given that manufacturing was the center of the mechanical economy that dominated the first three-quarters of the 20th century, this functional diagram looks just like a manufacturing company. The manufacturing company continued to serve as the model around which the six functional segments of enterprise management applications emerged, although with increasing variation depending on industry and specific company made continuously more economically feasible by the cost-constant annual growth in the transistor density of integrated circuits (ICs – aka “chips” such as CPU’s and other digital logic processors, memory, and later, solid-state storage), a rough proxy for computing capacity.

But we are getting ahead of ourselves. Prior to the invention of the first commercial transistor in 1954, the electronic computers of the late 1940s and mid-‘50s used vacuum tubes to keep “the state” of the machine and flip bits from 1 to 0 and back depending on the resulting electrical signal from the preceding series of vacuum tubes in a circuit. These early computers could only be programmed for a single application – essentially a series of computational tasks – and reprogramming required not only in-depth knowledge of the logical structure of the hardware but also knowledge of the physics of that hardware. While these computers still added substantial value in solving certain computationally challenging problems (e.g., modelling nuclear reactions), these systems were far too slow and expensive for almost all of the commercial applications that really started to take off in the ‘70s.

That began to change after the commercial availability of the transistor (1954). Yet, the first room-sized mainframes could still only muster a tiny fraction of the computing capacity of even 1980s era PCs. There was not even enough computing capacity to support an operating system, which handles and hides from the programmer routine low level tasks. So, initially, the software market was an aggregate of custom applications programmed from the ground up for each customer. The commercial availability of integrated circuits (ICs or “chips” as they are colloquially called), beginning in 1964, kicked off what became known as Moore’s Law, the cost-constant doubling of transistor density, a rough proxy for computing capacity, every two years until at least 2013 or until today (depending on whether one believes argument from Intel or from TSMC).

The ability of the nascent semiconductor industry to pattern and etch progressively smaller features on a chip enabled more and more of those transistors and their connecting circuitry onto wafers made mostly of silicon yielded chips with hundreds of transistors by the late ‘60s and thousands by the early ‘70s. In turn, this gave the likes of IBM, Sperry, Burroughs, and other computer vendors enough computational power for a room-sized mainframe to operate a low-level operating system “on top” of which a specific application could run.



Source: CFRA.

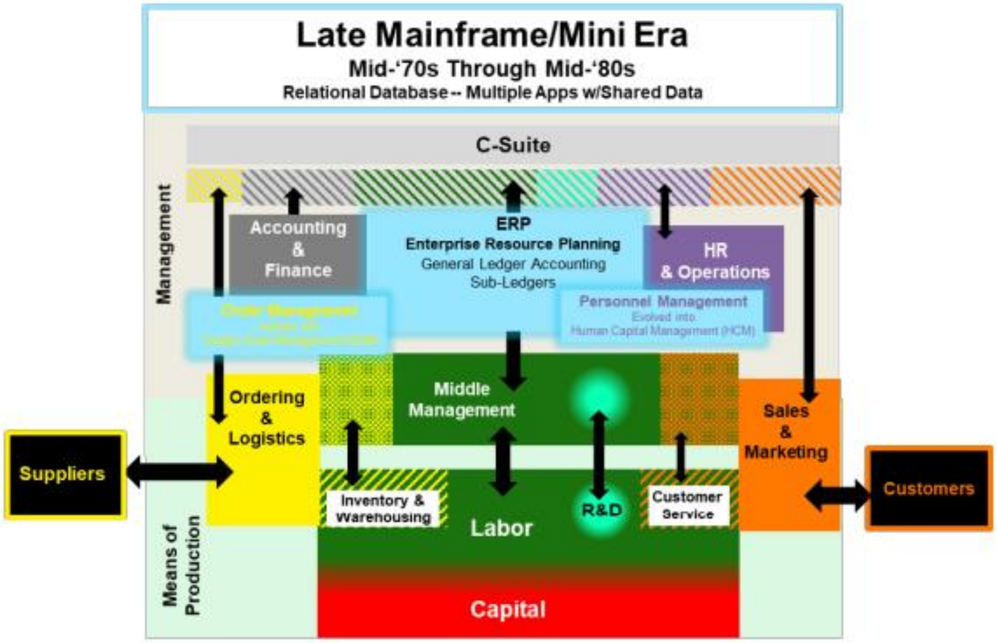
Mainframe operating systems, such as IBM's famous OS 360 that emerged in the mid-'60s, first thought of as an unnecessary performance-dragging luxury for programmers, became a standard feature for mainframes by the early '70s as the annual surplus of computing capacity delivered by Moore's Law made the performance overhead of operating systems increasingly infinitesimal by comparison. At the same time, Moore's Law multiplied the benefits of operating systems for programmers, saving considerable development time by handling a number of low-level tasks common to all applications and delivering greater value to customers. Further, operating systems allowed customers to upgrade their computer systems without having to rewrite their applications, preserving their original software investment and convincing them to increase that investment in the future.

Operating systems also made "time sharing" feasible, essentially allowing multiple applications to take turns accessing the computing capacity of a single mainframe, again adding to the return on investment (ROI) of software by enterprise customers. Among the most common business or "enterprise" applications that enabled organizations to attain "multinational" scale were those that helped maintain multiple tiers of accounting ledgers, automate the updating of entries from ledger to ledger, and facilitate faster, more frequent, and more accurate "closing the books." Other scale-enabling "business software" pictured above in the diagram, "Early Mainframe Era," automated calculations for payroll for hundreds of thousands of employees, invoices for tens of thousands of customers, and purchase orders for thousands of suppliers – all at different exchange rates and under different tax regimes depending on the country and local jurisdiction.

Generally, these applications were employed by the accounting and HR departments. By the mid-'70s, these applications had generated a significant amount of data and made it all accessible very quickly (versus manually retrieving information from filing cabinets), enabling enterprise organizations to create applications to forecast product demand, input costs, and pricing trends, which led to more quantitative determination of hiring and capital expenditure needs.

In fact, by the early '80s, one of the primary use cases for "enterprise software" was helping managers plan and optimize the mix of labor and capital in the means of production and is where we get the term "Enterprise Resource Planning," better known as ERP. German software company SAP specialized in this type of software, making it more applicable horizontally from company to company within the manufacturing industry during the '70s, and then from industry to industry in the '80s, expanding ERP's addressable market to large companies in most major industries. To this day, SAP is the global market share leader in ERP, through the market for ERP applications continued to divide and fragment both horizontally, becoming narrower by function, and vertically, becoming narrower by industry and sub-industry.

This process of division and specialization continues today. At first, in the late '70s, what had been two key ERP "modules" evolved into software categories in their own right: Human Capital Management (HCM), for tracking employee data and automating payroll calculation and distribution, a function previously accomplished by hand and infamously prone to errors; and Supply Chain Management (SCM). However, it should be noted that for many ERP vendors (including SAP), HCM and SCM remained as modules within their expanding ERP "suite." Before the '80s, enterprise software was run on mainframe systems and, later, "mini" computer systems (see diagram below). The output of these systems, in the form of highly structured reports, was typically only produced for upper-level management, especially within the HR and accounting departments to run payroll, track large numbers of personnel, record transactions, produce financial reports, periodically close the books, etc.

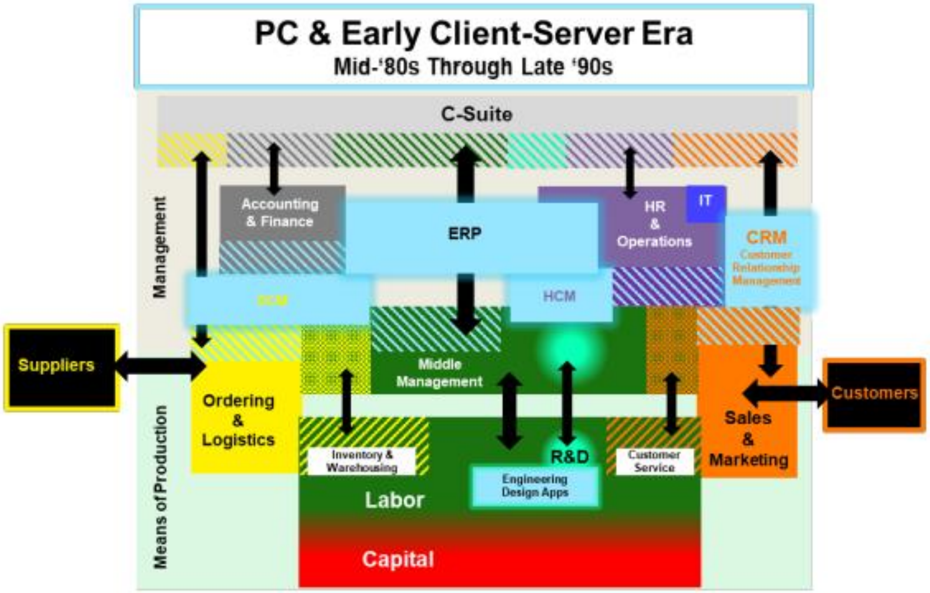


Source: CFRA.

That is why the first major categories of enterprise software applications were ERP, which included the accounting "system of record," HCM and SCM. The use of software to automate and improve management efficiency really began to take off in the mid-80s with the arrival of the IBM PC running Microsoft's DOS operating system, solidified by the early '90s as power shifted in PCs from IBM to the WinTel duopoly. In the '80s, generally speaking, managers – the so-called "white collar workers" – were the first people to get PCs on their desks with personal productivity applications, the emergence of corporate email, and access to centralized corporate applications running on mainframes and minis, later migrating to servers running the UNIX operating system, such as those from Sun Microsystems and HP, and finally to x86 servers running Windows NT or Linux.

However, even in the '90s, which saw the rise of the IT department as a functional organization separate from HR/Operations, those employed in roles traditionally described as "labor" or "blue collar workers" were rarely given PCs or access to software, especially not the general-purpose productivity-enhancing apps running on PCs. Released in 1992, version 3.1 of the Windows operating system is usually cited as the first version mature and stable enough for "real work." With its Graphical User Interface (GUI) clearly "inspired" by Apple's famously user-friendly Macintosh computers, Windows freed users from DOS's clunky and somewhat intimidating command-line interface, enabling easier and more intuitive navigation of a PC's operating system and its applications.

However, with Intel standardizing the motherboard and working with Microsoft to ensure optimal performance for Windows, the '90s saw an explosion of new entrants into the market for PCs, all running Windows, lowering the cost and making desktop PC standard issue for all levels of management across most industries. During the '90s, the sales and marketing functions got their "system of record" in the form of Customer Relationship Management (CRM) software, pioneered by Siebel System and pictured in the diagram below:



Source: CFRA.

Early "Digital Transformation"

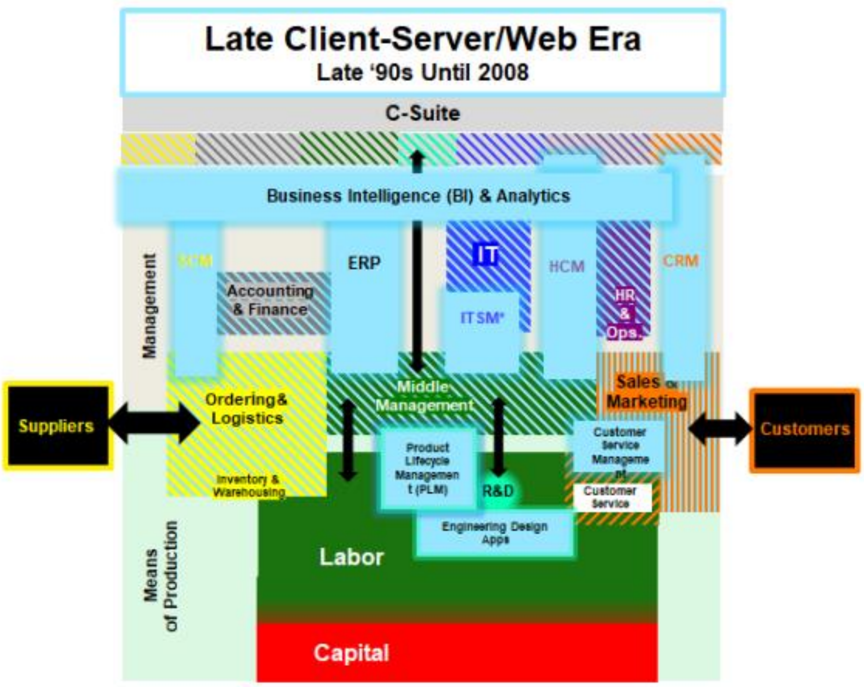
Even early on, there were several important exceptions to "the rule" that enterprise software was first used to aid and automate the management of people and capital. One example can be found in industrial process control systems featuring embedded software on the factory floor – very much the heart of the means of production. In the '80s and '90s, these systems evolved into simple mechanical robots that handled specific, repetitive functions in production lines, often substituting for labor, though with mixed results early on. These systems and their robotic descendants have a substantial hardware component – i.e., traditional, physical capital. Yet, the key aspect of what set them apart from the dumb machinery they replaced was their embedded software.

Two other important exceptions were the first Computer Aided Design (CAD) and Computer Aided Modelling (CAM – i.e., simulation software) applications that gained initial traction in the '70s, enabling line engineers and other technicians involved in product design and testing to tackle computationally challenging, "high pay-off" problems encountered with big ticket systems in aerospace/defense, architecture, and civil engineering. Despite these exceptions, enterprise software was, by in large, a tool of management. However, that began to change in the '80s with the PC and its more powerful, more expensive sibling, "the workstation," from the likes of HP, Silicon Graphics, and, most famously, Sun Microsystems.

Initially, these "means of production" apps used for design and simulation were custom-built, just like the initial apps for managing the means of production. The economic feasibility of desktop computing powerful enough to render precise 2D engineering diagrams and architectural blueprints gave rise to software companies dedicated to building apps for these "niche" use cases. Those companies are not only still around today but continue to dominate their respective segments, which are no longer niches, and are beginning to grow faster as cost-constant increases in computing capacity not only makes them more powerful in their capabilities but also much easier to use, expanding their respective user bases: Ansys for complex physics simulation, Autodesk for 2D architecture and engineering design, PTC for rapid prototyping in 3D, and Adobe for 2D creative design, photo postprocessing, and desktop publishing.

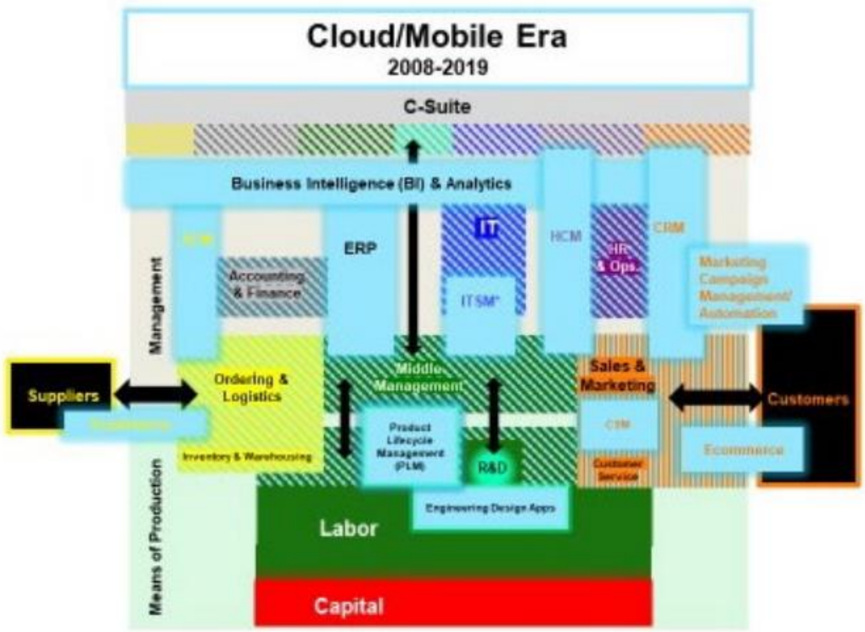
Further, with the proliferation of desktop and then mobile laptop/notebook PCs, IT departments expanded rapidly and began to require their own functional category of software, with IT Systems Management (ITSM) emerging as its own market segment in the late '90s with the final major functional category of the Client-Server Era. Business Intelligence (BI) apps took off in the early 2000s to gather, process, and analyze the data from the other five enterprise apps in order to summarize and present key metrics and insights from that data on "dashboards" to EVP-level and C-level executives who, by 2000, had PCs on their desks and were beginning to use email. Another key development at this time was the emerging use of applications offering a web-based interface, ostensibly making it unnecessary to load client software on the PC or other "client."

However, accessing corporate applications via a standard web browser – at least within large enterprises – did not really take off until cloud computing began to become viable for enterprises, not just small to medium-sized businesses (SMBs) that were cloud computing's first adopters prior to 2008. When the financial crisis struck in 2008, however, many large enterprises did make the move to cloud computing for certain specific apps, especially CRM, where the intrinsic mobility of salespeople gave the cloud model a strong advantage over the existing installed base of client-server CRM apps from the likes of SAP and Oracle.



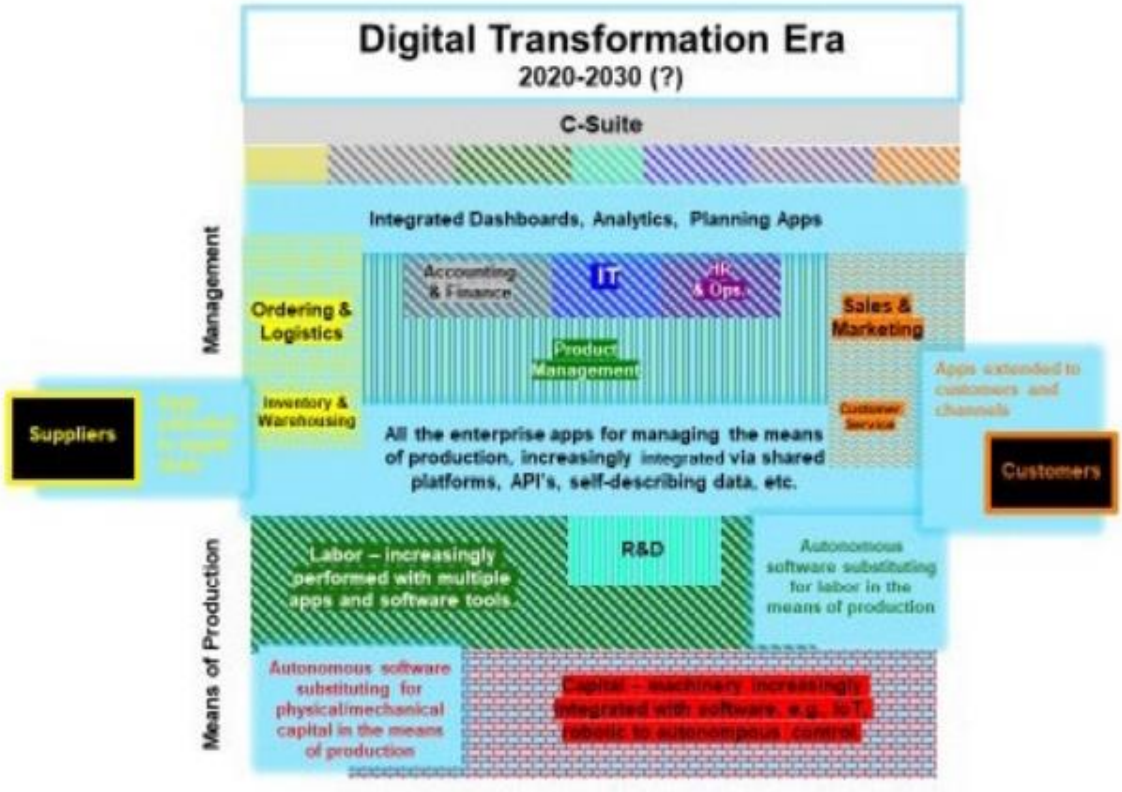
Source: CFRA.

From 2008, enterprise applications began moving to cloud computing more rapidly, although more mission critical apps at larger organizations, such as those in the ERP, SCM, and HCM categories, did not really migrate to cloud services until security and reliability concerns were assuaged beginning around 2015. The use of Amazon AWS to provide cloud infrastructure by emerging cloud app providers, and then subsequently by existing cloud app providers, contributed significantly to enterprise CIO's blessing migration to cloud computing.



Source: CFRA.

In our view, we have recently entered a new era, accelerated by the Covid-19 pandemic – Digital Transformation – although admittedly these days, with so many simultaneous trends and developments, clean demarcations of different "eras" is more difficult. Nevertheless, today, software has expanded well beyond its own industry into virtually every other industry by substituting for both labor and capital in the means of production in order to defend an established market position or to completely disrupt existing industries. Uber represents a clear example, effectively substituting software for the labor and capital required for taxi dispatching and, in so doing, was able to scale the dispatching function to a national level, then beyond, with code distributed within cloud data centers and among the apps downloaded to hundreds of millions of smartphones, enabling a widely distributed, self-serve model for riders and drivers leveraging the latest cellular mobile and GPS networks.



Source: CFRA.

RULE #7. GOLDMINERS AND SHOVEL SHOPS

There is an old investor aphorism that says, when experiencing a "gold rush" of opportunities in a particular area – like the dotcom era of the mid-to-late ‘90s or the literal bitcoin miners of the mid-to-late '10s – don't buy the stocks of the proverbial prospectors looking to get lucky in their pursuit of gold; rather, buy the stocks of their key suppliers – "the shovel shop." During the dotcom gold rush, the most important suppliers included Cisco, Ascend Communications, Dell, Sun Microsystems, Akamai, Juniper Networks, Redback Networks, and Oracle. One may have noticed a lean toward network equipment, which benefited most, earliest, as the infrastructure of the internet was initially constructed. However, revenues for many like Cisco were driven also by enterprise still building and upgrading their internal data networks. During the bitcoin gold rush, Nvidia, AMD, and Arista Networks benefited most. Generally, this advice held up during the gold rush period itself. But it's not a rule...

What if a company developed a pickaxe so powerful, with enough built-in intelligence to dig 1,000 times deeper and 100 times faster than competing equipment, with the ability to extract and sort all of the metals? Then, with a little more programming labor, what if it turned itself into an operating mine, transforming quickly from there into an entire ongoing mining operation, with multiple mines covering multiple geographies. And, what if the metal extracting power of this pickaxe-mine combo contraption doubles every two years? And what if the application of that power produces even greater earnings growth over time?

Can you imagine if Uber had tried to sell its software to taxicab companies instead of using it to become one? Actually, what Uber did was even better. In one of the largest and fastest cases of disintermediation ever, Uber used its software to strip out the taxi dispatch function from the capital heavy and labor-intensive transportation function, then make that function bidirectionally self-serve and scaled it nationwide. Could you see Booking.com selling its software to hotel operators or to travel agencies rather than using it to become the massive software-based hotel booking agent it has?

What would have happened if Mark Zuckerberg had licensed his social media software to media companies? It's quite possible he would have built a successful multi-million-dollar business, but he probably would not have created the world largest social network worth hundreds of billions of dollars. And, if evolution had taken the industry that way, as improbable as that seems sitting here now, we all might be members of "FoxFace Social," "MyMSNBCspace," or "Rupert and Lachlan's Endless Rave." Don't laugh. Crazier things have happened.

At this point, *all* outcomes feel highly improbable as they unfold, but none of them should be that surprising. And, sometimes, you build a tool that's just too good to sell to others.

Horsin' Around

Back in the '90s, everybody wanted to invest in "the internet," but not everyone had the risk appetite for dotcom stocks, especially when their business models looked vague, shaky, and dependent on too many assumptions about how they might, in the future, generate an operating profit, with some struggling to even generate revenue. Yet, the opportunities created by the internet were clearly very real and most could see the tremendous potential of the new medium. While most people do not put tech progress in these terms or talk about it in quite this way, following the Moore's Law and innovation-related frameworks discussed in Rules #2 to #4, one could project that in future states of transistor densities, there will be enough computing capacity per dollar to connect everyone seamlessly and instantly to what is effectively an endless source of information, music, shows, movies, video games, and any other content one could imagine.

Even before Marc Andreessen developed the first browser in '94, media and telecom companies were already publicly discussing the construction of an "Information Super Highway" and had described many of the types of services that such a highway would provide – online shopping, online print media, what they called "video-on-demand," video-conferencing, etc. – all of the software-based services available on the internet that we take for granted today. Admittedly, perhaps the foreknowledge among investors was not that specific, but it sure seemed like the internet was like a "gold rush of new business opportunities" based on networked software. Clearly, a few of these early companies would strike immense mountains of proverbial internet gold. Yet, even tech-savvy investors at the time had trouble telling the difference between those companies that were about to operationalize their discoveries and scale up lucrative mining operations and those that would end up bankrupt prospectors.

The gold rush analogy was as popular back then for the internet as bitcoin and the other blockchain-based coins are today. Taking a lesson from actual historical gold rushes, many investors followed the rule and avoided the stocks of dotcom companies, the prospectors in this modern-day gold rush. No one could be sure what business models would work, whether e-commerce would face resistance from potential buyers uncomfortable putting their credit card number online, or whether web sites could really sell enough advertising on the brand-new medium to cover the cost of acquiring all of those "eyeballs" without trashing the online experience of newcomers to the web. Chances are, as the logic of the gold rush went, only a fraction of the dotcoms would be successful, and it would be tough to tell which ones would survive, let alone thrive. In other words, whatever above average returns were reaped from investing in the gold miners, it was probably not worth the risk.

So, the prevailing wisdom among tech-savvy investors of the day remained through the decade: invest in those companies selling to the prospectors, i.e., the would-be gold miners. During the original narrative, one should invest in the vendors selling denim jeans, coffee, jerky, whiskey, salt, and all the other supplies to the gold miners. But the most important supplies were the gold miners' key tools for prospecting and mining. At the birth of the World Wide Web, when the internet went mainstream, considerable capex was invested by the dotcoms, the Internet Service Providers, and a variety of telcos and cablecos, which were given much greater freedom to compete in each other's spaces due to the 1996 Telecom Reform Act. That capex was directed at network equipment, servers, storage systems, and software – mostly infrastructure software, like operating systems and databases. Network equipment benefited most in the '90s as the initial internet infrastructure began to be built. Of course, we also saw a substantial surge in sales of PCs and peripherals as more individuals jumped on the internet and began surfing and exploring.

When the bubble burst and VC money for additional dotcom and telecom capex dried up, many dotcoms and alternative telecoms did not survive. This, in turn, hit the "shovel shop," though, for most, not as hard as the dotcom and telco "goldminers" because most shovel shops were also selling to the broader enterprise market. Hence, for those shovel shops, revenue and profitability held up better than the dotcoms looking for new gold on the internet. The operative word here is "most," and it didn't matter what category of business model was chosen. The vendors that did best were those that had a more diverse, enterprise customer base. Many shovel shops that were too reliant on the dotcoms and alternative telcos – even providing seller financing – did not survive (e.g., Nortel), or were acquired at a steep discount (Lucent), or never saw valuations as high again (e.g., Extreme Networks, JDS Uniphase, Juniper Networks). And, for a few goldminers that survived, like Amazon and Google, they turned out to be among the most lucrative long-term equity investments once the smoke cleared from the crash (although Google remained private until 2004).

By the mid-2000s, dotcom customer demand came back from the likes of Yahoo, Alphabet (formerly Google), Amazon, eBay, and, on the margin, the first wave of social media companies like Myspace and Meta (formerly Facebook) as well as cloud application providers like salesforce.com. All of these companies had hit their respective veins of internet gold and were now scaling up mining operations. However, because of the trail-off in capex by and subsequent orders from the dotcoms and the new entrants in network and telecom services, the shovel shops were now more beholden to the needs of their enterprise customers and were not especially interested in building products even for the now successful dotcom businesses. And the needs of enterprise customers were quite different from the more diverse and specific needs of the dotcoms and other internet gold miners. Enterprise needs were more universal and self-similar, thus encouraging the shovel shops to focus on those needs, rather than those of a handful of gold miners where even survival was in doubt.

The Rise of the Gold Miners

Meanwhile, Amazon and Google began to encounter scaling issues very specific to their software applications and their businesses. These issues were not being addressed by the shovel shop due to the fundamentally different nature of their requirements (e.g., problem of "scale-out" using systems and software designed to "scale up"). Amazon and Alphabet were solving multiple scalability problems across several dimensions, problems yet to be faced by even the largest, most mission-critical data centers operated by banks and a few large, software-centric retail chains like Walmart and CVS. And, given the already massive and still rapidly rising scale of Amazon's and Google's original, core software businesses, both were forced to adapt. Often, this adaptation meant "making it up as they went along." Both companies ended up custom-building multiple types of operating systems and databases from the ground up. Both even built their own custom development and monitoring tools internally. In effect, the gold miners had to build their own pickaxes and shovels.

Cloud Software Re-Engineering

While both companies were making unprecedented ground-up software development efforts to meet their unique and unprecedented requirements, they were also freed of many of the previous constraints and compromises that kept developers fenced in by the limitations of operating systems and databases that had to be engineered more for reliability and processor efficiency during previous states of much lower computing capacity per dollar. Many of these design constraints were put in place in the '70s and '80s, when hard drives and wide area network connections (i.e., links provided by telcos) were far less reliable, and had become redundant or moot by the 2000s, but they had permeated deeply into enterprise software architectures.

Mechanical hard drives were software's historical design constraint. In the '70s, due to the unreliability of hard drives and network connections, constraints had to be placed on how database software was structured, sacrificing flexibility of data organization for greater certainty, fidelity, security, and consistency of each database transaction. Any update to the data entailed a series of messages, receipt acknowledgments, handshakes, and other software processes that ensured the fidelity of the system, but that held back bigger jumps in innovation, even later on when these constraints had become moot with greater network reliability.

Many of these constraints are embodied in the “ACID principles”, which governed database transactions and were incorporated into databases supporting the SQL language and associated specifications. This had become the de facto standard for commercial databases through the '90s, despite continued work on alternative, non-SQL – which became "NoSQL" – database structures in academia and in open-source communities. Later, in states of much greater computing capacity, we saw NoSQL branch off into NOSQL or "Not Only SQL," databases, with structures flexible enough to support both "kinds" of databases.

Spillover
Many of the innovations both Amazon and Google made spilled over and eventually benefitted the overall market, not just their internally developed software. This "spillover" came initially as open-source contributions to more generally applicable development efforts in the open-source developer community, usually released a few years after they deployed similar code to solve their own very specific internal issues. For example, in 2008, largely due to the contributions of Google, Amazon, Facebook, and others, we saw a so-called "Cambrian Explosion" of a variety of open-source databases. Google has been a particularly prolific open-source contributor over the years, though, obviously, not for the purely altruistic motives that come to mind when one thinks of contributions to open source. Google's most successful open-source project was the Android mobile operating system, originally a defensive maneuver to keep Apple and, at the time, potentially Microsoft, Nokia, and Blackberry from establishing a foothold in smartphone operating systems. Google also established and was the primary contributor to the Hadoop family of open-source tools used to handle widely distributed databases and massive sets of unstructured, non-uniform data, as well as tools to analyze those data sets. Although, many ambitious initial promises of the "big data" movement have been left unfulfilled... at least at the present state of computational capacity per dollar.

Facebook too wrestled with hard drive-related scaling issues that had no precedent. In mid-2012, right as Facebook went public, users began taking a lot more pictures and videos with their now 8-megapixel (8MP) smartphones, then uploading and downloading these pictures and videos. Consequently, overall response times on the site became noticeably longer given that 90% of Facebook's data storage was on hard drives, which had increasingly become a glaring and problematic bottleneck within the hardware computing infrastructure of data centers everywhere.

However, Facebook's problem was especially acute, and its system architects struggled with user behavior that broke previous conventions. For example, assumptions based on the 80/20 rule, which had established such a great track record with respect to network traffic and content access, broke down as Facebook users were looking at all of the pictures posted by all of their friends, not just 20%. Facebook could get much fatter network links from the telcos and upgrade the servers with more memory, which it did. However, it would be wasted as hard drive transfer speed was *the* bottleneck and had become a major performance problem.

At that time, solid state storage (NAND Flash) featured transfer rates that were 2 to 4 orders of magnitude greater than hard drives but were still several times more expensive. However, it appeared that Facebook did not have a choice, so it invested several billions in flash-based storage. But Facebook went well beyond that, designing its own storage systems, even re-writing the low-level drivers for those flash drives in order to maximize performance and eliminate many of the delay-inducing "hand-shake" messages and other compromises related to mechanically driven hard drives that permeated the system at multiple levels.

Yet, unlike the two other dotcom gold miners, Amazon and Google, Facebook has yet to build such a pickaxe business from the "spillover" of their respective software development efforts. Ostensibly, Facebook's Oculus Rift VR goggles and development platform would qualify as a pickaxe vendor business model but selling VR goggles does not take advantage of Facebook's hyperscale cloud operation. Besides, Oculus Rift generates less than 2% of Facebook's revenue. This analyst remains puzzled as to why Facebook has yet to enter the cloud infrastructure service business.

So, what does this all mean for investors today? We draw three conclusions.

First, neither the gold miner nor the pickaxe vendor category of the business model for software-based businesses is inherently superior to the other. Both models offer examples of success and above average returns for investors over the long term since the beginning of the software industry. Two "OG" gold miners from the dotcom era, Amazon and Alphabet, have certainly generated enough return over the last two and a half decades to prove a glaring exception to the rule of always going with the shovel shop in a gold rush. Certainly, the rule appears to hold for the two digital anecdotes – dotcom and bitcoin – and likely held historically for physical gold rushes, but primarily during that initial "rush" period. Or one could argue that once the rush is over, the criteria changes and one can find better opportunities among established miners that have struck a vein. At that point, one is betting on growth and duration. And, it must be said that Amazon and Alphabet have continued to maintain 20%+ annual top-line growth even now.

It should be noted that this dichotomy is as old as software. For example, Fair Isaac Corporation (FICO), which has operated in both modes for over half a century, originally used a gold miner model, which uses its internally operated software to generate its FICO credit scores that are now 50% of revenue and more than 100% of net profit. ADP and its "software-automated payroll calculation service" is one of oldest examples of a pre-web "OG" gold mining software-based business.

Second, *during* a "gold rush" period, investing in the stocks of companies following a pickaxe vendor model, as opposed to investing in the gold miners, appears to be a better risk-reward strategy. For us today, a gold rush is that period of time following a fundamental breakthrough in software – breakthroughs fueled by the geometric jumps in computing capacity per dollar that unlock some large increment of value. These breakthroughs are usually confined to a certain area but can sometimes be deeply foundational, the stuff of multi-decade above-average returns.

For example, when computing capacity hit the price point and form factor to have a programmable computer running third-party apps on every manager's desk, which was a crucial breakthrough that cascaded into the mainstreaming of several software apps that have added incalculable value through personal productivity – word processing, spreadsheets, relational databases, apps for graphics, typography, and page layout for desktop publishing. This is still sometimes referred to as "the PC gold rush of the 1980s," but the opportunity was almost exclusively for shovel shops, with Microsoft and Adobe among the stocks delivering "many X" returns since. In addition, it pays to bet on those shovel shops that have diverse customer bases – the ones that will see only slower growth for a period if a gold rush evaporates, not steep absolute declines in their businesses.

Of course, none of these breakthroughs are independent; they are, in fact, closely linked and intensely iterative within the realm of software, which is a big reason why software-based gold rushes, from the limited to the large, are happening all of the time now – fintech, medtech, edtech, headtech – all of the companies in these emerging opportunity spaces are all based on cloud software. Yet, to date, the biggest single breakthrough in terms of incremental value created is still likely the internet, specifically the World Wide Web, which gave birth to such an expansive variety of software-based activities that are now a part of the daily experience for most of humanity. The web remains a deeply layered and expanding platform of software with millions of apps, games, and other types of software all running on platforms on top of platforms on top of platforms. Aside from the invention of the transistor and the integrated circuit, the arrival of the web and commercial internet was perhaps the biggest most value-enabling development in the evolution of software and the software industry. It fundamentally changed how software is developed and began to knock down the barriers between the two modes of business, a process more active today than ever.

Certainly, the "rush" part of the gold rush is as much feeling as it is reality. If the opportunities are very real, they entail potential risk, and therefore also entail a substantial psychological component. Investors in software-based businesses *feel* a gold rush phenomenon when mainstream investors realize that software has hit some breakthrough. The rush happens when certain assumptions about the world start going "poof" in rapid sequence and things thought possible but not yet feasible become downright lucrative looking, resulting in what feels like an unending domino-like cascade of innovation.

Yet, to any prudent investor, that rush feeling should also induce some healthy fear of loss, especially in the beginning, before the nuts and bolts of the research and valuation work are completed and can validate the size of the alleged mineral deposit, or the reality of the opportunity. In our next "rule," we'll discuss some of the clues in "Differentiating the Cons from the Benioffs." Suffice to say, while it is necessary to frequently employ abstractions and metaphors when discussing software and the software industry, beware of the use of software jargon that seems abstracted from *all* reality, including the reality of software.

Our third and perhaps most important conclusion is that the internet, cloud computing, 5G, and the ongoing geometric growth in computing capacity are converging to increase the fluidity with which software-based businesses can move between pickaxe vendor and gold miner models. We believe it is something to keep in mind as cloud computing fully replaces client-server in the enterprise over the next several years. To date, this crossover was mainly a byproduct of scale – the scale of cloud computing infrastructure, to be specific, with Microsoft, Amazon, and Alphabet all operating businesses in both categories of models. However, we also see fluidity between both categories growing without being a result of scale. Certainly, the enterprise cloud app providers, such as salesforce.com, ServiceNow, and Atlassian, will be much larger than they are today, according to our forecasts (and even according to consensus). They will be looking for new growth vectors. Remember, salesforce.com tried to buy social media company LinkedIn, only to be outbid by the largest pickaxe vendor, Microsoft.

As an aside, if one *really* believes in the opportunities of a particular gold rush, it might be better not to invest in the stocks of gold miners but to *become* a gold miner yourself. After all, micro fortunes are being made around the world by those developing apps and games in both categories of business models, while gaining narrow but very appreciative customer bases.

Cloud-based companies do inherently possess much greater agility in adjusting their business model, in our view. They can, and frequently do, expand into both business model categories. Therefore, when analyzing the fundamentals and projecting the financials, one should keep a more open mind regarding where the opportunities and threats may come from. This is not a surprise from a big picture perspective but is a conclusion that needs to be taken to heart when evaluating the stocks of any software-based business. Since cloud computing and its increasingly associated Micro-Services Architecture (MSA) remove much of the friction from automating data exchanges and establishing linkages among apps, games, and more fundamental, commonly shared software services and infrastructure, it creates a much more fluid threat and opportunity matrix; thus, there are more adjacent market segments than one thinks or than one can track. The good news from a competitive standpoint is that equilibrium can be better maintained through constant renewal, as cloud software, especially that is based on an MSA model and employing Agile development methodologies, is continually updated to adapt to market conditions, competitive threats, and emerging opportunities. Therefore, the cycle of innovation for cloud-based software businesses is faster than non-software businesses but also faster than that of traditional software companies.

All cloud-based software businesses already share a DevSecOps culture. Cloud-based software businesses are already handling one of the most challenging aspects of a cloud-based model: operating networked software at scale. Well, to be precise, "developing-operating" networked software at scale, as successful cloud companies require both core competencies – software development (dev) and operations (ops) to act in tandem. Then, let's add security or "sec" so that we now evaluate a cloud-based software business in terms of their "DevSecOps" core competencies. Security competence is critical in order to scale and grow; no matter how silly an app or video game may seem, it's not silly to somebody if it hits a million daily active users (DAUs), and it's not silly to anybody when it hits 10 million DAUs. Generally, whether gold miner or pickaxe vendor, cloud-based software companies look more like each other than like traditional software companies, possessing much greater overall operational fluidity and the ability to invade adjacent market segments and, with the skyrocketing number of interconnections and API's, there are so many more market segments that are operationally adjacent to cloud-based software providers and ripe for invasion.

Also, for those opportunities requiring other necessary core competencies or possessing a particularly valuable and tough-to-penetrate customer base, acquisition is increasingly attractive. Again, this has much to do with the considerable reduction in the friction of integration that cloud-based software enables and is especially true in a strong DevSecOps culture with a cloud-first approach built under an MSA model and with a surplus of well-maintained Application Programming Interfaces (APIs). In contrast, traditional, license-based enterprise software deployed on customers' premises usually required years to build integrations between existing apps and the acquired code, a synergy-promising task that was just as often abandoned after the deal closed.

In terms of internal development and organic expansion, the biggest moves have been from gold miner to pickaxe vendor, the most notable being Amazon's expansion into cloud infrastructure services with AWS and Google's expansion with Google Cloud, as we have discussed. While smaller in terms of direct revenue, we remember that Apple, which had closely followed a pickaxe vendor model since its inception, made an incursion into gold mining a little over 20 years ago when it moved from being just a seller of devices with its embedded software (iPod) to a direct toll-taker and more importantly the price-setter ($0.99/song) within the music industry (iTunes). This occurred once it was possible to make the lion's share of music's long tail available with the cloud, or via Apple's proprietary network connection tunneled across the internet. We also remember that this incursion paved the way for the iPhone and Apple's present trillion-dollar valuation.

Microsoft, another close follower of the pickaxe vendor model, began making moves into gold mining, such as the launch of its internet search engine, Bing, paid for by advertising revenue just as its more dominant rival Google. While Bing remains a distant #2 to Google in the search advertising business, its launch helped justify the large necessary investments in data center capacity and cloud infrastructure software, enabling Microsoft to mount a defense against AWS. That defense became Azure, the not so distant #2 cloud infrastructure/platform service responsible for much of Microsoft’s recent growth reacceleration.

We are also seeing signs of more business model fluidity in the pricing schemes of cloud-based shovel shop that capture more upside potential from their customers. For example, Twilio (TWLO) provides a catalog of free APIs and development tools that allow developers to insert a wide variety of real-time voice, video, or messaging capabilities into their apps in what is close to a plug-and-play basis. But TWLO also maintains the cloud software invoked via those APIs. That software sets up the necessary network connections, handles the messaging on the back end, and coordinates all of the exchanges of session data and telecom billing information to the appropriate parties. TWLO does not transport the network traffic itself; it deals only with the data *about* the network traffic.

The plummeting cost of transporting network traffic, which has already fallen so precipitously (in most countries) thanks to Moore's Law, was already encouraging gratuitous consumption of real time communications and messaging. However, building that functionality into software applications is not easy and requires expertise in specialized and somewhat arcane telecom-related protocols and software, not a particularly common or easily acquired skill set for app or game developers. TWLO APIs are activated any time a user clicks a button or when a player's avatar gets let into a VIP room at a digital furry convention, prompting a broadcast message for friends to join. After a certain number of messages or transactions or communications sessions, depending on the API, TWLO then begins charging on a per usage basis – per message, per session, per minute, per connection, etc. This enabled TWLO to ride the success of Uber (forgive the pun) after Uber's developers used the TWLO API for users to click and call or text their driver and communicate anonymously through the app.

RULE #8. DIFFERENTIATING THE CONS FROM THE BENIOFFS

So far in our "rules," we have discussed things to look for or, at least, we have described some places along the riverbank where bigger, better, multi-X-return candidates can be found. However, fish, along with IR departments, are not dumb. They will notice what is being noticed by investors and the adage of "fake it 'til you make it" becomes more of something to uncover and less of something to inspire beginners.

All companies cover their bases, meaning that they strive to have answers for what they see as the questions they are most likely to get. Also, know that someone else will ask the obvious questions. Whether they are asked publicly by the sell-side or privately by the buy-side, the questions usually fall within a fairly narrow range, and much of what is asked is informational, the kind of question best left to email. This is why, if one has the chance to ask questions directly to management, surprise them. Ask about the company, not the results. Ask about how they see the industry, not their company's position in it. Has their recruiting strategy changed? Ask about what software tools the company uses internally.

Thar's info gold in 'dem conversational tangents! We think it is understandable but nonetheless insane that analysts ask questions they know the CxO will not or cannot answer. Often, in response to the question, "Why did you ask a question you knew the CEO would not answer?" they sometimes claim that they were listening to the CEO's "tone" when she answered a question to figure out what the "hidden truth" is. Even if that was the genuine intention of the analyst, that answer is lame, in our view. This analyst has never been able to discern anything except annoyance from a CEO's tone when she retorts, "We haven't disclosed that." Analysts ask unanswerable questions to which we all want the answers because they: 1) are lazy but want to look eager; 2) are attempting to publicly align themselves with misguided investor rage by putting a CxO on the spot; 3) lack the imagination or rhetorical skills to ask questions that elicit good, informational, unscripted responses; and 4) assume that management knows the answer and is hiding it.

Of course, one should not go overboard in constructing gotcha questions to assess the health of the corporate culture of a company one is researching. For example, one should not see whether a CEO means what he says about a tightknit culture by asking him, "Your receptionist, Steve, of whom you spoke so fondly, what's his last name?". That might end awkwardly. Even after almost two decades, one might really regret asking such questions, that one time, many years ago, just to gain a shred of investment-relevant insight.

These are more general pieces of advice in sizing up a management team, but fundamental research into software-based businesses must also contend with the inherently fuzzy and abstracted language of software, the evolution of which is fueled by what seems like an informational avalanche of trends, sub-trends, buzz words, and other linguistic cattle prods that can go viral quicker than an X-coin "finfluencer" on TikTok. With terminology that both changes so rapidly and represents ever more abstract concepts, like cloud, virtual machines, AI, containers, etc., it can be difficult to figure out who's fakin' and who's makin'.

It's not that those abstract concepts aren't real. They are very real. As fuzzy as they sound, "cloud" and "metaverse" are real phenomena. Rather, it is that their abstracted meaning in the context of software makes them susceptible to manipulation, turning them into tools of exaggeration, obfuscation, and time-buying until the fruition of some plan that management is still figuring out. As investors, we need to be aware of verbal stalling or misdirection tactics in the service of keeping a stock price elevated. In the world of jargon-heavy software and software-based businesses, these situations can be especially difficult to discern.

One does not have to be an industry expert to be a successful investor, but without a cognitively helpful framework and a solid high-level understanding of software and the software industry, one is forced to rely solely on stories told by company executives, IR departments, industry analysts, sell-side analysts, and *even* independent investment research analysts, all of whom have their own biases, perspectives, and investment horizons. Even if it is simply the task of obtaining accurate market-research, background information on the relevant technologies, or comparisons with competitors, relying solely on sources not in complete alignment with one's investment objectives can be dangerous and unconducive to generating alpha.

To be clear, in this rule, we are not talking about actual "cons" or fraudulent behavior (although one should not rule it out if the evidence is there), or even any behavior for which the company would be civilly liable. IR rhetoric need not rise to even close to that level in order to be damaging to investors. Conversely, there is a lot to be said for competent, effective IR departments. Good ones can single-handedly provide explanations for certain dynamics in the industry that cannot be readily had through other sources. However, invariably, an effective IR department is the result of a strong management team and solid operational execution within a healthy, well-maintained corporate culture. So, ironically, good IR is often wasted on well-managed companies.

Regardless of IR's competence, we believe it is critical for investors to view a company's IR as *a* source, not *the* source. Often, it is in the best interest of IR to spread out good news over time and, to the degree practical, set the expectations bar low enough to avoid misses and disappointments. Often, the ability to sense great news brimming behind a CFOs' knowing smirk is more helpful to alpha generation than recognizing "that stutter" when a CEO is trying to suppress news of a big competitive loss and blaming the quarter on the sudden appearance of seasonality. Whether or not one is in a position to ask questions of company management or whether or not one is an institutional investor, listening to earnings calls, not just reading the transcripts, can be very helpful in enabling one to discern key changes in tone and thus gain some degree of insight vs. those not paying as much attention.

Despite previous disparaging of "tone readers" as a sub-field of equity research, the way CxOs talk can tell a lot. We have found it helpful to gauge a CxO's tone along with *changes* in word choice and lingo, though primarily when she or he is talking *outside* of the normal parameters of short-term investment considerations. However, *watching* members of management speak and field questions at conferences, in person or on video, especially industry conferences, is even better. Noticing management's visual and verbal patterns over time can be critical to understand the present dynamics of a company relative to its past. Seeing and listening also helps in knowing management's communication style and therefore the ability to spot deviations from or changes to that style. As one might expect, the unrehearsed responses to questions are much more helpful than the content of any pre-written script.

It's just a turn of phrase until you start unconsciously parroting IR. Now that we are on the topic of interacting with IR departments, CEOs, and management generally, we have one fairly specific recommendation. Actually, it is more of a request. Whether one is on the buy-side, sell-side, investor-side – regardless of one's role – please stop asking, "How should *we* think about 'topic X'?" We understand that it is simply meant to be a question format to prompt more open-ended discussion from management about "Topic X". Though we find the phraseology mildly annoying, it's no worse than any other expression that has jumped the shark, become tired, and no longer needs quotation marks. All viral terminology in the investment industry and in "business-speak" more generally – phrases like, "at the end of the day" or "moves the needle" – eventually stop moving the needle at the end of the day. The problem is deeper.

Beware the cognitive vampires. In the view of this analyst, with all the cognitive biases that we now know bedevil humans trying to make investment decisions, it just does not seem like a good idea to tempt the power of subconscious suggestion, which is a very real and measurable phenomenon. After all, this job is hard enough as it is. In the opinion of this analyst, questions of the form "how should we think about..." are reckless invitations wander around unchaperoned inside one's cognition factory, exploring the warehouses that store all of one's raw opinions, soon-to-be insights, and WIP analysis. That access should not be given to anyone, let alone someone as biased as the CxO of the company being researched and analyzed. Questions of the form "how should we think about..." are the analytical equivalent of letting the Lost Boys, Nosferatu, and Dracula come right through your open, garlic-free front doorinto your home to join you for some midnight BBQ.

Moreover, that question and those like it are bad for our industry. They epitomize intellectual laziness and make us all look and sound bad. This analyst implores his fellow analysts – peers, allies, and competitors alike – please do not literally invite IR to indoctrinate you. At least, please, don't do it in public in full view of your customers. Maybe there's a program.

One does not need to be an expert or a software industry guru to invest successfully in the equities of software-based businesses. Far from it – but it's helpful to be able to recognize puffery from confidence. And, given the large, unique personalities of those running many of today's software companies, one's normal investor instincts and usually accurate nonsense detection techniques might fail under a hail of jargon and strawman customer anecdotes that often feel like verbal 3-card monte with clouds, platforms, and "proprietary" algorithms – now playing at a metaverse near you.

Interestingly, if one read the previous paragraph and then listened to salesforce.com CEO Marc Benioff back, say, in 2007, just before Saleforce.com really began expanding from its core small to medium-sized business (SMB) market into larger enterprises for its cloud-based Customer Relationship Management (CRM) application, one may have seriously doubted the story Benioff was telling. And, if one had doubted him, one would have been dead wrong.

To effectively catch opportunities like the one SF.com presented in 2007, investors need to make other judgments at the company level. In fact, this analyst likes to employ a scoring system to help analyze and compare the company-specific fundamentals of software-based businesses over the years. More on this scoring system later in "the rules." Yet, before employing any scoring system, one can save some time by filtering out the inevitable wanna-be "NewSoft" companies and those stretching their metaphors too far with their "We're the Uber of X" schtick, which often occurs when they realize their original schtick just isn't big enough and soon they won't be able to hide their slowing growth. So, management starts talking about "Act 2," peppering everything with a bunch of new jargon, and one starts to question if that jargon holds any real meaning.

Many institutional investors were wrong at the time about Benioff and salesforce.com. More specifically, they doubted salesforce.com's prospects for convincing large enterprises to abandon the investments most had made in client-server CRM solutions from market share leaders at the time, Oracle (Siebel), SAP, and Microsoft, to a Software-as-a-Service (SaaS) solution where the data and the code are hosted by an upstart provider and accessed over the internet via a web browser. However, that is exactly what happened, propelling salesforce.com to #1 in CRM market share within five years. Yet, at the time, how could one tell the difference?

At the time, we noticed five ways one could tell that salesforce.com was what Benioff said it was and what it was likely to become outside of the normal research pathways, channel checks, and market research. However, we will admit that the more precise articulation of these five signs took longer to fully formulate. Clever anecdotes cannot form rules, but they can tell what data to check.

First, Benioff could explain in detail the value proposition of salesforce.com's approach in practical terms without explaining his jargon by using more jargon. Benioff had experience as a programmer and therefore was quite familiar with the technical concepts he was discussing. However, his experience at Oracle made him also intimate with the sales process for enterprise software, making his strategy more believable, if one were at least somewhat familiar with that process. It also legitimized his product expertise. If Benioff had founded the cloud company to revolutionize an area of enterprise software with which he was unfamiliar, like the Supply Chain Management (SCM) market segment, his declarations about the problems customers of existing SCM solutions were facing would have carried less weight, in our view.

Second, in response to more detailed questions about how salesforce.com was able to provide a secure, scalable, customizable, easy-to-use CRM app at a level of performance and reliability suitable for the largest enterprise, Benioff would explain – as much as was practically possible – how salesforce.com accomplished this without any fear of spilling "corporate secrets." He knew that salesforce.com's secrets were deeply embedded in the software code and in the human procedures that co-evolve with that code. So, Benioff had no problem giving investors unfettered – albeit practically time-restricted – access to the company's CTO, Parker Harris, to effectively verify and fortify Benioff's claims and explanations. Both Parker and Benioff never claimed that they could not tell you the answer because it was proprietary to salesforce.com. They would thoroughly explain the next level of understanding; your ability to understand what they were saying was up to you. They understood the vast difference between a detailed, abstracted explanation to a sophisticated investor and handing that investor all of the source code and operational procedures.

If the management of any software-based business cannot or is reluctant to explain the guts of how their solution works because they claim it would risk their competitive advantage by "giving away the goods," it probably means that there might not be any "goods" to give away in the first place. It might be that they are afraid to admit that what they do isn't that hard or proprietary after all. Worse, they might actually believe their own propaganda. On the other hand, Benioff and Parker loved to describe in greater detail how their solution was differentiated, as if they could not be prouder of the details of what they built and liked to rub it in the face of their competitors.

Third, Benioff also embraced operational transparency in a refreshingly direct and honest way by building a "Trust" section of the salesforce.com site that showed the real-time status of and performance stats for the system. In this way, the CIOs of larger enterprises considering making the shift to salesforce.com could compare their own internal uptime and performance stats with that of salesforce.com. If salesforce.com's uptime, performance, and security-related stats exceeded that of a particular large enterprise, which was invariably the case, it kind of destroys the objection that migrating the internally operated CRM app to the web-based salesforce.com would be risky in term of maintaining the existing level of uptime, performance, and security parameters.

Well-informed confidence is contagious. Benioff was confident that his team, working full-time to keep its software app up and running and secure, could match or exceed the uptime and performance stats of virtually any large enterprise IT department responsible for keeping hundreds of apps and dozens of mission critical apps up and running. More importantly, he had and continues to have no problem demonstrating that confidence.

Fourth, Benioff talked exactly like his customers. Benioff might be a special case in that the large majority of salesforce.com users in 2007 were salespeople or sales support staff employed at SMBs, individual proprietorships, and, increasingly at the time, salespeople at large companies using the app as individuals. In the context of salesforce.com's customers, even the hyper-enthusiastic, super salesy approach Benioff seems to adopt all of the time – even on earnings calls to this day – starts to make sense. At the time, he wasn't selling to the IT department, at least not at first; he was selling to salespeople and doing so in the language and style with which they are most familiar.

Fifth, if one interviewed salesforce.com's customers at the time, which this analyst did extensively at the time, one realized that they not only endorsed the app, but their enthusiasm also matched Benioff's! From the beginning, salesforce.com customers didn't want to be just users, they wanted to become experts in using salesforce.com; they wanted to differentiate themselves and gain a leg up in their careers by integrating salesforce.com into their sales and account management processes as deeply as possible. The salesforce.com CRM application had become a critical cognitive tool for "hired gun" individual salespeople like real estate agents, talent agents, insurance brokers, and a growing number of unaffiliated, self-selling professionals of all stripes. Every year, a portion of these sales professionals would land jobs at larger companies, "metastasizing" from an SMB or hired gun/contractor role. In many cases when this occurred, they brought their salesforce.com subscription and all of the associated data (i.e., contact information, correspondence history, etc.) with them, continuing to use the tool with which they had grown so comfortable and competent.

With individual salespeople at large enterprises using salesforce.com and neglecting to use the legacy client-server solution maintained and controlled by the IT department, salesforce.com's Trojan Horse strategy would at least get them in the door and in the mix. This was despite frequent opposition by enterprise IT personnel who were often initially skeptical that SaaS was "enterprise grade." It was harder for IT to make that argument when over half the users were already using and strongly preferring salesforce.com to the legacy CRM app.

As positively enthusiastic as these early users were, they had lots of critiques and suggestions too, and they were clearly paying attention to and looking forward to new features, greater scalability, and a greater ability to customize the solution. This told you two things: 1) there is already waiting demand for what the software might be able to do in future states of Tr/mm2/$ (i.e., greater computing capacity per unit cost), which is not always the case; and 2) users were already benefitting a lot by becoming expert users of the salesforce.com application, blending their cognition with, for lack of a better term, software's "cognition."

APPENDIX

Recommended Glossaries:

Note: Rather than maintain a glossary in this publication, we have decided to evaluate and recommend several online sources we think are particularly helpful, well maintained, frequently updated, and accessible to non-tech, generalist readers while also being definitive and accurate. We did reject a number of sources that were poorly maintained, included long-winded, overly technical, or unclear definitions. We caution that many sources that come up in online searches for "tech glossaries" or "software terminology" are very poorly maintained or woefully incomplete, serving only as clickbait to drive traffic to commercial sites, which are usually selling materials or services for software training, education, certifications, etc. We will continue to add to this list, especially for lists focused on specific areas within software that we think might be relevant and helpful to investors.

gartner.com – Perhaps the most comprehensive list of terms relevant to the software industry with clear, carefully worded definitions, understandable by those without a background in tech. It is oriented toward enterprise software and may not have the latest hacker jargon, more informal tech slang, or terms found outside of the world of enterprise software that is Gartner's focus. Nevertheless, as the world's largest research and analysis firm specifically covering the technology industry, the company plays a role not only in presenting the state of the world in tech, but also in influencing and, even to some extent, defining that state of the world.

techopedia.com – A very broad glossary of all things tech. Despite the broad coverage, technopedia.com is very well maintained and updated daily. Its definitions are accurate, generally well written, and accessible and helpful to generalists.

The Ultimate Tech Glossary for Recruiters and HR Managers – While this document comes in a static pdf format and it's unclear how often it is updated, we really liked the way complex terms are defined using simple, straight-forward language while maintaining overall accuracy. Further, the document is limited to about 100 of the most common terms, meaning that a person could simply read the entire document as a way to gain general familiarity.

techterms.com – One of the most comprehensive online glossaries – more of a dictionary in fact – though still accessible to and written for non-technical people. One nice feature on the front page: a list of the most popular and recently entered terms, giving a sense of what is trending.

whatis.techtarget.com – This is another very comprehensive list and includes many terms outside of the realm of software. The definitions are written in a more formal and technical style but appear quite accurate when we browsed the entries. The site also provides a dashboard with a breakdown of terms by subject category, helpful if one wants to get up to speed in a particular area of software.

INDUSTRY REFERENCES

PERIODICALS

CNET News
cnet.com
Guide to news, information, and events related to technology companies.

Computerworld
computerworld.com
Covers computer hardware and software.

CRN
crn.com
Covers computer hardware and Software industry news.

eWeek
eweek.com
Covers news and developments in e-business, communications, and internet-based architecture.

InformationWeek
informationweek.com
Provides news and features on the computer hardware and software industries.

InfoWorld
infoworld.com
Covers computer hardware and software.

PC Magazine
pcmag.com
Covers news in the personal computer industry.

Software Magazine
softwaremag.com
Covers the software industry, with a focus on information technology managers.

GOVERNMENT AGENCIES

U.S. Bureau of Economic Analysis
bea.gov
Produces and disseminates statistics that provide a comprehensive, up-to-date picture of U.S. economic activity.

TRADE ASSOCIATIONS AND RESEARCH FIRMS

Forrester Research Inc.
forrester.com
Analyzes and predicts the impact of technological changes on large companies, consumers, and society.

Gartner Inc.
gartner.com/technology/home.jsp
Researches and analyzes developments and trends in the IT industry. Its Dataquest unit serves IT suppliers and the financial and investment communities.

Grand View Research
grandviewresearch.com
Researches and analyzes developments and trends across a range of industries including chemicals, materials, energy, health care, and technology.

IDC, Inc.
idc.com
The leading market research firm for enterprise software, cloud services, computing hardware, and enterprise network equipment sales.

The Software & Information Industry Association
siia.net
Provides data and services related to the software publishing industry.

ONLINE RESOURCES

BSA | The Software Alliance
bsa.org
Provides resources to combat software piracy.

Internet World Stats
internetworldstats.com
Provides data regarding internet users.

Net Applications
netapplications.com
Provides data regarding internet usage and market share statistics.

RightScale
rightscale.com
Conducts surveys on the adoption of cloud computing.

StatCounter
statcounter.com
A web traffic analysis website used to compute web usage share.

COMPARATIVE COMPANY ANALYSIS

					Operating Revenues															
					Million $							CAGR (%)			Index Basis (2013=100)					
Ticker		Company		Yr. End	2021	2020	2019	2018	2017	2016	2015	10-Yr.	5-Yr.	1-Yr.	2021	2020	2019	2018	2017	2016
APPLICATION SOFTWARE																				
ADBE	[]	ADOBE INC.		DEC	15,785.0	12,868.0	11,171.0	9,030.0	7,301.5	5,854.4	4,795.5	14.1	21.9	22.7	329	268	233	188	152	122
ANSS	[]	ANSYS, INC.		DEC	1,906.7	1,681.3	1,515.9	1,293.6	1,095.3	988.5	942.8	10.7	14.0	13.4	202	178	161	137	116	105
ADSK	[]	AUTODESK, INC.	#	JAN	4,386.4	3,790.4	3,274.3	2,569.8	2,056.6	2,031.0	2,504.1	6.9	8.6	15.8	175	151	131	103	82	81
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	3,435.0	2,988.2	2,336.3	2,138.0	1,943.0	1,816.1	1,816.1	10.0	10.5	11.4	189	165	129	118	107	100
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	1,024.2	842.5	824.1	740.7	676.2	630.1	611.6	NA	10.2	21.6	167	138	135	121	111	103
INTU	[]	INTUIT INC.		JUL	9,633.0	7,679.0	6,784.0	6,025.0	5,196.0	4,694.0	4,192.0	10.8	15.5	25.4	230	183	162	144	124	112
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	1,055.5	841.4	737.7	566.3	433.0	329.1	224.7	33.8	26.2	25.4	470	375	328	252	193	147
PTC	†	PTC INC.		SEP	1,807.2	1,458.4	1,255.6	1,241.8	1,164.0	1,140.5	1,255.2	4.5	9.6	23.9	144	116	100	99	93	91
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	5,777.8	4,854.2	4,727.7	5,191.2	4,607.5	3,789.9	3,582.4	7.5	8.8	19.0	161	136	132	145	129	106
CRM	[]	SALESFORCE, INC.	#	JAN	26,492.0	21,252.0	17,098.0	13,282.0	10,540.0	8,437.0	6,667.2	29.1	26.1	24.3	397	319	256	199	158	127
SNPS	[]	SYNOPSYS, INC.		OCT	4,204.2	3,685.3	3,360.7	3,121.1	2,724.9	2,422.5	2,242.2	10.6	11.7	14.1	188	164	150	139	122	108
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	1,592.3	1,116.7	1,086.4	935.3	840.9	759.9	591.0	17.8	15.9	42.6	269	189	184	158	142	129
SYSTEMS SOFTWARE																				
FTNT	[]	FORTINET, INC.		DEC	3,342.2	2,594.4	2,163.0	1,804.6	1,494.9	1,275.4	1,009.3	22.7	22.7	22.7	331	257	214	179	148	126
GEN	[]	GEN DIGITAL INC.	#	APR	2,796.0	2,551.0	2,490.0	2,456.0	2,559.0	4,019.0	3,600.0	-8.5	-8.5	-8.5	78	71	69	68	71	112
MSFT	[]	MICROSOFT CORPORATION		JUN	168,088.0	143,015.0	125,843.0	110,360.0	96,571.0	91,154.0	93,580.0	9.2	9.2	9.2	180	153	134	118	103	97
ORCL	[]	ORACLE CORPORATION	#	MAY	42,440.0	40,479.0	39,068.0	39,506.0	39,383.0	37,792.0	37,047.0	1.3	1.3	1.3	115	109	105	107	106	102
NOW	[]	SERVICENOW, INC.		DEC	5,896.0	4,519.0	3,460.0	2,608.8	1,918.5	1,391.0	1,005.5	46.7	46.7	46.7	586	449	344	259	191	138

Note: Data as originally reported. CAGR-Compound annual growth rate.
[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the following calendar year.
Souce: S&P Capital IQ.

Net Income

Ticker		Company		Yr. End	Million $ 2021	2020	2019	2018	2017	2016	2015	CAGR (%) 10-Yr.	5-Yr.	1-Yr.	Index Basis (2013=100) 2021	2020	2019	2018	2017	2016
APPLICATION SOFTWARE																				
ADBE	[]	ADOBE INC.		DEC	4,822.0	5,260.0	2,951.0	2,591.0	1,694.0	1,168.8	629.6	19.2	32.8	NM	766	836	469	412	269	186
ANSS	[]	ANSYS, INC.		DEC	454.6	433.9	451.3	419.4	259.3	265.6	252.5	9.7	11.3	NM	180	172	179	166	103	105
ADSK	[]	AUTODESK, INC.	#	JAN	497.0	1,208.2	214.5	-80.8	-566.9	-582.1	-330.5	19.0	NM	NM	-150	-366	-65	24	172	176
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	785.0	696.0	989.0	345.8	204.1	203.1	203.1	25.4	27.9	NM	387	343	487	170	100	100
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	-75.4	-4.0	78.7	-60.6	3.3	-79.4	-104.7	NA	-1.0	NM	72	4	-75	58	-3	76
INTU	[]	INTUIT INC.		JUL	2,062.0	1,826.0	1,557.0	1,329.0	985.0	979.0	365.0	12.5	16.1	NM	565	500	427	364	270	268
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	196.0	143.5	180.6	137.1	123.5	70.4	20.9	72.3	22.7	NM	936	685	862	654	590	336
PTC	†	PTC INC.		SEP	476.9	130.7	-27.5	52.0	6.2	-54.5	47.6	18.8	NM	NM	1003	275	-58	109	13	-115
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	1,152.6	949.7	1,767.9	944.4	971.8	658.6	696.1	10.4	11.8	NM	166	136	254	136	140	95
CRM	[]	SALESFORCE, INC.	#	JAN	1,444.0	4,072.0	126.0	1,110.0	360.0	323.0	-47.4	51.4	NM	NM	NM	NM	-266	NM	-759	-681
SNPS	[]	SYNOPSYS, INC.		OCT	757.5	664.3	532.4	432.5	136.6	266.8	225.9	13.1	23.2	NM	335	294	236	191	60	118
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	161.5	194.8	146.5	147.5	169.6	113.7	64.9	19.3	7.3	NM	249	300	226	227	261	175
SYSTEMS SOFTWARE														NM						
FTNT	[]	FORTINET, INC.		DEC	606.8	488.5	331.7	334.9	31.4	32.2	8.0	25.5	79.9	NM	7597	6116	4153	4193	393	403
GEN	[]	GEN DIGITAL INC.	#	APR	836.0	554.0	3,887.0	31.0	1,138.0	-106.0	2,488.0	-0.7	-25.9	NM	34	22	156	1	46	-4
MSFT	[]	MICROSOFT CORPORATION		JUN	61,271.0	44,281.0	39,240.0	16,571.0	25,489.0	20,539.0	12,193.0	10.2	24.4	NM	503	363	322	136	209	168
ORCL	[]	ORACLE CORPORATION	#	MAY	6,717.0	13,746.0	10,135.0	11,083.0	3,587.0	9,452.0	8,901.0	4.9	9.1	NM	75	154	114	125	40	106
NOW	[]	SERVICENOW, INC.		DEC	230.0	119.0	627.0	-26.7	-116.8	-414.2	-198.4	NA	NM	NM	-116	-60	-316	13	59	209

[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the following calendar year.
Souce: S&P Capital IQ.

Ticker		Company		Yr. End	Return on Revenues (%) 2021	2020	2019	2018	2017	2016	Return on Assets (%) 2021	2020	2019	2018	2017	2016	Return on Equity (%) 2021	2020	2019	2018	2017	2016
APPLICATION SOFTWARE																						
ADBE	[]	ADOBE INC.		DEC	30.5	40.9	26.4	28.7	23.2	20.0	17.7	21.7	14.2	13.8	11.7	9.2	34.4	44.2	29.7	29.1	21.3	16.2
ANSS	[]	ANSYS, INC.		DEC	23.8	25.8	29.8	32.4	23.7	26.9	7.2	7.3	9.3	12.8	8.8	9.5	10.6	11.5	14.8	17.1	11.6	12.1
ADSK	[]	AUTODESK, INC.	#	JAN	11.3	31.9	6.6	NM	NM	NM	5.8	16.6	3.5	NM	NM	NM	54.8	292.4	NM	NM	NM	NM
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	0.0	23.3	22.0	42.3	16.2	10.5	NA	15.9	15.0	29.5	14.0	8.4	0.0	26.6	25.7	58.3	30.4	23.6
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	NM	NM	9.5	NM	0.5	NM	NM	NM	1.3	NM	0.0	NM	NM	NM	4.5	NM	0.7	NM
INTU	[]	INTUIT INC.		JUL	21.4	23.8	23.0	22.1	19.0	20.9	13.3	16.7	24.8	25.9	24.2	23.0	27.5	41.2	47.4	63.7	78.3	46.1
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	18.6	17.0	24.5	24.2	28.5	21.4	6.1	5.5	7.3	9.0	8.0	6.5	25.3	24.3	41.9	44.5	62.1	65.6
PTC	†	PTC INC.		SEP	26.4	9.0	NM	4.2	0.5	NM	10.6	3.9	NM	2.2	0.3	NM	27.4	9.9	NM	5.9	0.7	NM
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	19.9	19.6	37.4	18.2	21.1	17.4	4.9	4.0	9.8	6.2	6.8	4.6	8.9	8.3	19.0	12.9	15.4	11.9
CRM	[]	SALESFORCE, INC.	#	JAN	5.5	19.2	0.7	8.4	3.4	3.8	1.5	6.1	0.2	3.6	1.6	1.8	2.9	10.8	0.5	8.5	4.0	5.2
SNPS	[]	SYNOPSYS, INC.		OCT	18.0	18.0	15.8	13.9	5.0	11.0	8.7	8.3	8.3	7.0	2.5	5.1	14.8	14.7	14.1	12.8	4.2	8.4
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	10.1	17.4	13.5	15.8	20.2	15.0	3.4	7.5	6.7	8.2	10.5	8.4	7.5	10.8	10.0	11.7	16.1	12.8
SYSTEMS SOFTWARE																						
FTNT	[]	FORTINET, INC.		DEC	18.2	18.8	15.3	18.6	2.1	2.5	10.3	12.1	8.6	10.9	1.4	1.5	73.3	44.4	28.2	41.9	4.4	4.0
GEN	[]	GEN DIGITAL INC.	#	APR	29.9	21.7	156.1	1.3	44.5	NM	12.0	8.7	50.3	0.2	7.2	NM	NM	NM	20.1	NM	22.7	NM
MSFT	[]	MICROSOFT CORPORATION		JUN	36.5	31.0	31.2	15.0	26.4	22.5	18.4	14.7	13.7	6.4	10.2	10.6	47.1	40.1	42.4	19.4	31.9	27.0
ORCL	[]	ORACLE CORPORATION	#	MAY	15.8	34.0	25.9	28.1	9.1	25.0	6.1	10.5	8.8	10.2	2.6	7.0	7,301.1	147.3	57.8	32.0	7.1	18.5
NOW	[]	SERVICENOW, INC.		DEC	3.9	2.6	18.1	NM	NM	NM	2.1	1.4	10.4	NM	NM	NM	7.0	4.8	38.7	NM	NM	NM

Note: Data as originally reported. CAGR-Compound annual growth rate.
[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the following calendar year.
Souce: S&P Capital IQ.

					Current Ratio						Debt/Capital Ratio (%)						Debt as a % of Net Working Capital					
Ticker		Company		Yr. End	2021	2020	2019	2018	2017	2016	2021	2020	2019	2018	2017	2016	2021	2020	2019	2018	2017	2016
APPLICATION SOFTWARE																						
ADBE	[]	ADOBE INC.		DEC	1.3	1.5	0.8	1.1	2.1	2.1	21.8	23.7	8.6	30.6	18.2	20.3	237.4	156.3	-58.3	743.7	50.6	62.5
ANSS	[]	ANSYS, INC.		DEC	2.1	2.4	2.2	2.5	2.1	2.2	14.4	16.3	10.9	0.0	0.0	0.0	87.6	80.6	49.2	0.0	0.0	0.0
ADSK	[]	AUTODESK, INC.	#	JAN	0.7	0.8	0.8	0.7	0.9	1.1	72.8	62.9	109.3	111.2	119.2	59.8	NM	NM	NM	NM	NM	396.8
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	0.0	1.8	1.9	1.7	1.3	1.5	NA	11.3	12.2	14.1	27.3	44.6	NA	46.7	50.9	69.6	183.9	216.1
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	1.1	1.1	1.1	1.1	1.0	1.1	33.4	23.7	25.9	29.1	48.0	53.1	277.5	252.1	261.1	403.3	620.4	581.5
INTU	[]	INTUIT INC.		JUL	1.9	2.3	1.8	1.4	0.7	0.7	17.1	28.5	9.3	12.1	24.4	60.6	81.3	45.6	23.7	57.1	-82.8	NM
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	1.1	1.1	1.1	1.0	1.0	1.0	3.0	4.3	5.5	8.9	11.1	19.8	10.5	19.9	26.2	90.7	96.8	107.6
PTC	†	PTC INC.		SEP	1.4	1.2	1.2	0.9	1.0	1.0	41.4	41.1	35.8	42.4	44.6	47.1	490.1	658.4	463.2	NM	NM	NM
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	0.8	0.7	0.8	1.1	0.9	1.2	38.1	46.4	33.0	39.0	38.8	50.1	NM	NM	NM	3,040.1	NM	1,753.6
CRM	[]	SALESFORCE, INC.	#	JAN	1.0	1.2	1.1	0.9	1.0	0.8	15.4	6.1	7.3	16.9	6.3	21.1	997.4	64.2	239.1	NM	NM	NM
SNPS	[]	SYNOPSYS, INC.		OCT	1.2	1.2	1.0	0.7	1.0	1.0	0.5	2.0	2.9	12.6	3.9	6.4	6.4	24.6	NM	-81.5	195.8	10,291.2
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	1.2	2.0	1.3	1.2	1.4	0.8	36.1	0.0	0.0	0.0	0.0	1.1	972.5	0.0	0.0	0.0	0.0	-12.7
SYSTEMS SOFTWARE																						
FTNT	[]	FORTINET, INC.		DEC	1.6	1.5	1.9	1.8	1.7	1.9	55.3	0.0	0.0	0.0	0.0	0.0	77.1	0.0	0.0	0.0	0.0	0.0
GEN	[]	GEN DIGITAL INC.	#	APR	0.7	0.7	1.2	0.8	1.1	1.2	103.5	117.9	99.7	40.8	50.0	66.4	NM	NM	758.2	NM	1,556.0	979.5
MSFT	[]	MICROSOFT CORPORATION		JUN	2.1	2.5	2.5	2.9	2.9	2.4	26.1	33.5	39.4	46.6	52.0	47.8	52.3	54.4	62.8	65.0	79.6	66.9
ORCL	[]	ORACLE CORPORATION	#	MAY	1.6	2.3	3.0	2.5	4.0	3.1	108.7	92.7	84.5	69.9	54.6	47.1	594.9	242.0	198.2	187.0	98.7	96.0
NOW	[]	SERVICENOW, INC.		DEC	1.1	1.2	1.0	1.2	1.2	1.3	28.7	36.7	24.6	37.3	44.7	56.8	547.6	208.9	937.7	199.2	164.7	187.4

Note: Data as originally reported. CAGR-Compound annual growth rate.

[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the following calendar year.

Souce: S&P Capital IQ.

Ticker		Company		Yr. End	P/E Ratio (High-Low) 2020	2019	2018	2017	2016	2015
APPLICATION SOFTWARE										
ADBE	[]	ADOBE INC.		NOV	49 - 26	51 - 34	52 - 32	54 - 29	47 - 31	73 - 55
ANSS	[]	ANSYS, INC.		DEC	72 - 40	48 - 26	38 - 27	50 - 30	32 - 27	35 - 28
ADSK	[]	AUTODESK, INC.	#	JAN	204 - 143	NM - NM	NM - NM	NM - NM	NM - NM	172 - 125
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	35 - 21	13 - 10	36 - 20	36 - 25	32 - 24	22 - 16
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	NM - NM	122 - 60	NM - NM	NA - NA	NA - NA	NA - NA
INTU	[]	INTUIT INC.		JUL	44 - 28	47 - 31	42 - 26	37 - 27	31 - 21	83 - 61
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	187 - 66	89 - 37	69 - 34	40 - 21	43 - 19	125 - 63
PTC	†	PTC INC.		SEP	86 - 43	NM - NM	238 - 125	1105 - 801	NM - NM	103 - 77
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	50 - 28	23 - 15	34 - 27	28 - 19	30 - 24	28 - 21
CRM	[]	SALESFORCE, INC.	#	JAN	1219 - 919	109 - 70	226 - 155	178 - 115	NM - NM	NM - NM
SNPS	[]	SYNOPSYS, INC.		OCT	52 - 25	41 - 22	35 - 28	96 - 63	35 - 23	36 - 27
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	95 - 54	79 - 47	65 - 45	40 - 31	58 - 38	95 - 55
SYSTEMS SOFTWARE										
FTNT	[]	FORTINET, INC.		DEC	51 - 25	56 - 34	47 - 22	250 - 167	199 - 128	1041 - 623
GEN	[]	GEN DIGITAL INC.	#	APR	4 - 3	595 - 360	18 - 14	NM - NM	7 - 4	21 - 16
MSFT	[]	MICROSOFT CORPORATION		JUN	35 - 23	27 - 18	48 - 32	22 - 15	22 - 16	33 - 27
ORCL	[]	ORACLE CORPORATION	#	MAY	19 - 13	18 - 14	61 - 51	20 - 17	21 - 16	20 - 17
NOW	[]	SERVICENOW, INC.		DEC	918 - 405	90 - 50	NM - NM	NM - NM	NM - NM	NM - NM

Ticker	Dividend Payout Ratio (%) 2021	2020	2019	2018	2017	2016	Dividend Yield (High-Low, %) 2020	2019	2018	2017	2016	2015
APPLICATION SOFTWARE												
ADBE	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
ANSS	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
ADSK	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
CDNS	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
CDAY	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
INTU	31.3	30.7	32.2	30.6	35.8	32.5	0.8 - 0.5	1.1 - 0.7	1.0 - 0.7	1.2 - 0.7	1.3 - 1.0	1.5 - 0.9
PAYC	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
PTC	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
ROP	20.5	22.5	10.8	18.0	14.7	18.4	0.6 - 0.5	0.8 - 0.5	0.7 - 0.5	0.6 - 0.5	0.8 - 0.5	0.8 - 0.5
CRM	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
SNPS	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
TYL	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
SYSTEMS SOFTWARE												
FTNT	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
GEN	36.2	67.3	6.5	700.0	18.5	NM	2.8 - 2.1	3.0 - 1.2	1.7 - 1.0	1.2 - 0.9	3.6 - 1.0	3.6 - 2.3
MSFT	27.0	34.2	35.2	76.6	46.5	53.6	1.1 - 0.9	1.5 - 1.1	2.0 - 1.4	2.3 - 1.7	3.0 - 2.2	3.3 - 2.5
ORCL	51.5	22.3	30.3	26.5	87.5	27.8	1.9 - 1.3	2.4 - 1.6	1.8 - 1.4	1.7 - 1.4	1.8 - 1.4	1.8 - 1.3
NOW	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0

Note: Data as originally reported. CAGR-Compound annual grow th rate.
[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the follow ing calendar year.
Souce: S&P Capital IQ.

Ticker		Company		Yr. End	Earnings per Share ($) 2021	2020	2019	2018	2017	2016	Tangible Book Value per Share ($) 2021	2020	2019	2018	2017	2016
APPLICATION SOFTWARE																
ADBE	[]	ADOBE INC.		DEC	10.0	10.8	6.0	5.2	3.4	2.3	0.7	2.4	-3.9	-6.7	4.6	3.2
ANSS	[]	ANSYS, INC.		DEC	5.2	5.0	5.3	4.9	3.0	3.0	3.6	4.2	6.6	10.4	8.4	8.2
ADSK	[]	AUTODESK, INC.	#	JAN	2.2	5.4	1.0	-0.4	-2.6	-2.6	-14.9	-8.8	-12.7	-13.4	-8.8	-4.2
CDNS	[]	CADENCE DESIGN SYSTEMS, INC.	#	JAN	0.0	2.5	2.1	3.5	1.2	0.7	0.0	5.7	5.4	4.5	1.4	0.2
CDAY	†	CERIDIAN HCM HOLDING INC.		DEC	-0.5	0.0	0.5	-0.6	-0.3	-1.4	-2.8	-0.9	-1.9	-3.6	-18.1	-20.8
INTU	[]	INTUIT INC.		JUL	7.6	6.9	5.9	5.1	3.8	3.7	3.7	13.1	7.8	4.4	0.1	-0.6
PAYC	[]	PAYCOM SOFTWARE, INC.		DEC	3.4	2.5	3.1	2.3	2.1	1.2	14.0	10.5	8.2	4.9	4.0	1.1
PTC	†	PTC INC.		SEP	4.0	1.1	-0.2	0.4	0.1	-0.5	-4.6	-3.7	-1.8	-4.3	-4.8	-5.5
ROP	[]	ROPER TECHNOLOGIES, INC.		DEC	10.8	9.0	16.8	9.1	9.4	6.4	-86.4	-101.6	-57.6	-52.7	-53.0	-64.1
CRM	[]	SALESFORCE, INC.	#	JAN	1.5	4.4	0.2	1.4	0.5	0.5	1.2	12.0	4.5	0.3	2.9	-1.5
SNPS	[]	SYNOPSYS, INC.		OCT	4.8	4.3	3.5	2.8	0.9	1.7	9.4	8.4	4.2	-0.2	2.1	2.7
TYL	[]	TYLER TECHNOLOGIES, INC.		DEC	3.8	4.7	3.7	3.7	4.3	2.9	-27.0	20.1	10.1	7.7	8.0	-0.1
SYSTEMS SOFTWARE																
FTNT	[]	FORTINET, INC.		DEC	0.7	0.6	0.4	0.4	0.0	0.0	0.7	0.9	1.4	1.1	0.7	0.9
GEN	[]	GEN DIGITAL INC.	#	APR	1.4	0.9	6.0	0.0	1.7	-0.2	-6.9	-7.7	-6.2	3.0	-9.5	-13.4
MSFT	[]	MICROSOFT CORPORATION		JUN	8.1	5.8	5.1	2.1	3.3	2.6	11.2	9.0	6.9	5.1	5.5	6.5
ORCL	[]	ORACLE CORPORATION	#	MAY	2.4	4.6	3.1	3.0	0.9	2.2	-19.3	-14.6	-11.6	-8.1	-1.0	0.8
NOW	[]	SERVICENOW, INC.		DEC	1.1	0.6	3.2	-0.2	-0.7	-2.5	13.2	12.5	9.6	4.8	3.2	1.4

Ticker	Share Price (High-Low, $) 2020	2019	2018	2017	2016	2015
APPLICATION SOFTWARE						
ADBE	536.9 - 255.1	332.9 - 215.2	277.6 - 175.3	186.3 - 102.8	111.1 - 71.3	96.4 - 69.0
ANSS	369.8 - 200.1	260.1 - 137.1	190.5 - 136.8	155.1 - 91.9	99.0 - 80.5	98.4 - 78.8
ADSK	307.2 - 125.4	185.8 - 121.1	159.9 - 101.6	131.1 - 74.5	83.1 - 41.6	65.8 - 42.1
CDNS	136.8 - 51.4	77.1 - 41.4	47.4 - 35.5	45.6 - 25.2	28.0 - 18.3	23.3 - 16.5
CDAY	111.9 - 38.4	68.6 - 33.2	45.0 - 28.7	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
INTU	387.9 - 187.7	295.8 - 188.2	231.8 - 150.4	161.4 - 111.9	118.7 - 88.2	109.2 - 79.6
PAYC	471.1 - 163.4	280.0 - 115.0	164.1 - 79.2	86.1 - 42.5	52.9 - 22.4	46.4 - 22.6
PTC	121.8 - 43.9	102.5 - 62.1	107.4 - 60.5	67.1 - 45.9	49.9 - 27.1	42.8 - 30.5
ROP	455.7 - 240.0	385.5 - 257.9	312.7 - 245.6	267.8 - 183.7	189.4 - 155.8	195.9 - 144.6
CRM	284.5 - 115.3	167.6 - 130.1	161.2 - 102.3	109.2 - 69.0	84.5 - 52.6	82.9 - 55.0
SNPS	261.1 - 104.9	146.7 - 81.0	103.4 - 79.1	94.8 - 59.0	62.0 - 39.3	52.8 - 41.6
TYL	466.2 - 247.2	301.4 - 176.3	252.5 - 173.3	188.2 - 142.8	175.8 - 118.2	184.0 - 103.2
SYSTEMS SOFTWARE						
FTNT	30.7 - 14.0	22.1 - 13.0	18.9 - 8.7	9.1 - 6.0	7.5 - 4.6	10.1 - 5.7
GEN	28.7 - 15.1	26.7 - 17.5	29.7 - 17.4	34.2 - 23.8	25.7 - 16.1	27.3 - 19.1
MSFT	232.9 - 132.5	159.6 - 97.2	116.2 - 83.8	87.5 - 62.0	64.1 - 48.0	56.9 - 39.7
ORCL	66.2 - 39.7	60.5 - 44.4	53.5 - 42.4	53.1 - 38.3	42.0 - 33.1	45.3 - 35.1
NOW	566.7 - 238.9	303.2 - 168.5	206.3 - 130.1	131.3 - 74.6	89.8 - 46.0	91.3 - 62.6

Note: Data as originally reported. CAGR-Compound annual grow th rate.
[]Company included in the S&P 500. †Company included in the S&P MidCap 400. §Company included in the S&P SmallCap 600. #Of the follow ing calendar year.
Souce: S&P Capital IQ.

Disclosures

CFRA's Industry Surveys Reports (the "Industry Surveys") have been prepared by Accounting Research & Analytics, LLC and/or one of its affiliates. The Industry Surveys are published and distributed by Accounting Research & Analytics, LLC d/b/a CFRA with the following exceptions: In the European Union/European Economic Area, the Industry Surveys are published and distributed by CFRA UK Limited (company number 08456139 registered in England & Wales with its registered office address at New Derwent House, 69-73 Theobalds Road, London, WC1X 8TA, United Kingdom), which is regulated by the UK Financial Conduct Authority (No. 775151); in Malaysia, the Industry Surveys are published and distributed by CFRA MY Sdn Bhd (Company No. 683377-A) and regulated by Securities Commission Malaysia, No. CMSL/A0181/2007 ("CFRA Malaysia"). CFRA Malaysia is a wholly-owned subsidiary of CFRA US. For Recipients in Canada: This report is not prepared subject to Canadian disclosure requirements and may not be suitable for Canadian investors.

The content of this report and the opinions expressed herein are those of CFRA based upon publicly-available information that CFRA believes to be reliable and the opinions are subject to change without notice. This analysis has not been submitted to, nor received approval from, the United States Securities and Exchange Commission or any other regulatory body. While CFRA exercised due care in compiling this analysis, CFRA AND ALL RELATED ENTITIES SPECIFICALLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, to the full extent permitted by law, regarding the accuracy, completeness, or usefulness of this information and assumes no liability with respect to the consequences of relying on this information for investment or other purposes.

No content (including ratings, credit-related analyses and data, valuations, model, software or other application or output therefrom) or any part thereof (Content) may be modified, reverse engineered, reproduced or distributed in any form by any means, or stored in a database or retrieval system, without the prior written permission of CFRA. The Content shall not be used for any unlawful or unauthorized purposes. CFRA and any third-party providers, as well as their directors, officers, shareholders, employees or agents do not guarantee the accuracy, completeness, timeliness or availability of the Content.

Past performance is not necessarily indicative of future results. This document may contain forward-looking statements or forecasts; such forecasts are not a reliable indicator of future performance.

This report is not intended to, and does not, constitute an offer or solicitation to buy and sell securities or engage in any investment activity. This report is for informational purposes only. Recommendations in this report are not made with respect to any particular investor or type of investor. Securities, financial instruments or strategies mentioned herein may not be suitable for all investors and this material is not intended for any specific investor and does not take into account an investor's particular investment objectives, financial situations or needs. Before acting on any recommendation in this material, you should consider whether it is suitable for your particular circumstances and, if necessary, seek professional advice.

CFRA may license certain intellectual property or provide services to, or otherwise have a business relationship with, certain issuers of securities that are the subject of CFRA research reports, including exchange-traded investments whose investment objective is to substantially replicate the returns of a proprietary index of CFRA. In cases where CFRA is paid fees that are tied to the amount of assets invested in a fund or the volume of trading activity in a fund, investment in the fund may result in CFRA receiving compensation in addition to the subscription fees or other compensation for services rendered by CFRA, however, no part of CFRA's compensation for services is tied to any recommendation or rating. Additional information on a subject company may be available upon request.

CFRA's financial data provider is S&P Global Market Intelligence. THIS DOCUMENT CONTAINS COPYRIGHTED AND TRADE SECRET MATERIAL DISTRIBUTED UNDER LICENSE FROM S&P GLOBAL MARKET INTELLIGENCE. FOR RECIPIENT'S INTERNAL USE ONLY.

The Global Industry Classification Standard (GICS®) was developed by and/or is the exclusive property of MSCI, Inc. and S&P Global Market Intelligence. GICS is a service mark of MSCI and S&P Global Market Intelligence and has been licensed for use by CFRA.