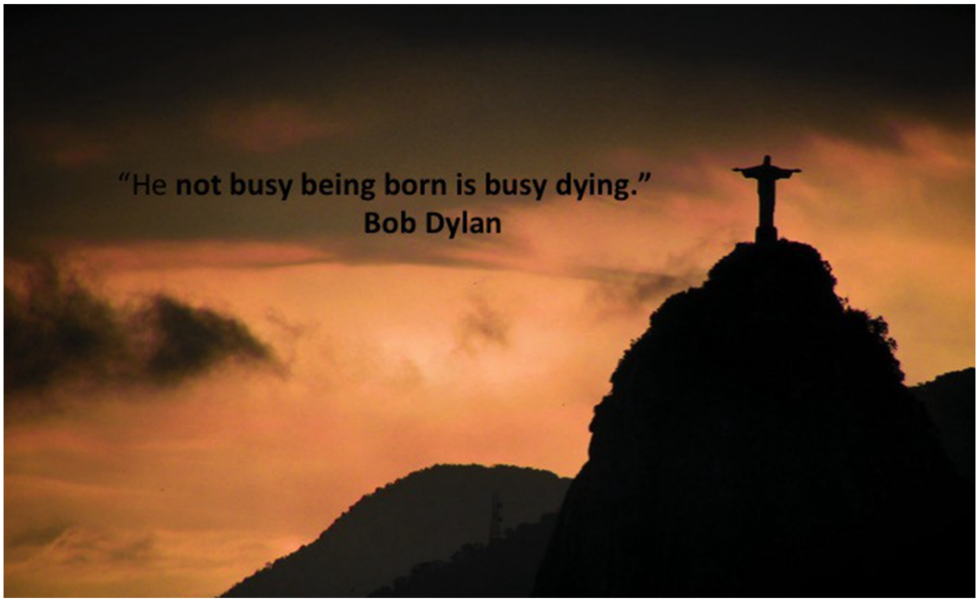

"He not busy being born is busy dying."—Bob Dylan. Photo: Anthony Goto

ENTERPRISE SOFTWARE: DEATH AND TRANSFIGURATION

PUBLISHED ON MEDIUM ON OCTOBER 23, 2016

Once upon a time—and it was a time that lasted some thirty years—there was no better place for VCs to invest in the broad world of tech than enterprise software. This is no longer true, and the enterprise is missing out as a result. What's an entrepreneur or VC to do?

This paper tracks the parallel revolutions in technologies and markets that have transformed the industry. The evolution of the enterprise software industry has compelling messages for multiple constituencies:

For venture investors and the entrepreneurs whom they back—the principal audience for this article—the economics of investing in enterprise are utterly and dauntingly different.
For enterprise customers, extreme consolidation of legacy ERP platforms and applications must drive strategies both to limit dependence on the dominant incumbents, Oracle and SAP, and simultaneously to bring operational coherence to the accelerating proliferation of innovative point solutions.
For public market investors, the dominance of Oracle ($37 billion in revenues and valued at 5 times revenues and 20 times EPS) and SAP ($23 billion in revenues and valued at more than 5 times revenues and 27 times EPS) is clearly recognized and will required continuous evaluation as the industry's radical evolution proceeds.

SOME HISTORY

To begin at the beginning, several factors contributed to the historic attractiveness of enterprise software as an arena for entrepreneurial investment:

A succession of standard platforms established a stable target environment: first, IBM's System 360 architecture; then, the canonical client/server architecture of smart Windows clients interacting with relational database servers running UNIX; and on to the fully distributed, web architecture with the application server written in Java at its core.
A second factor was the persistent and exponential decline in hardware costs, enabling value to migrate upward through higher and higher levels of abstraction, broadening the scope for algorithms to capture and automate a seemingly limitless range of business processes, from the first generation of general ledgers to comprehensive ERP systems to specialized processes unique to particular subsectors like banking or logistics.
Third, an external and internal ecosystem of customer-supporting systems management and application development resources evolved to build and deploy software to automate customer-specific processes: systems integration firms became less important as corporate IT

departments matured with a vastly increased supply of software engineers.

So enterprise application and enterprise infrastructure software companies co-evolved in symbiosis, the latter enabling the frontier for standard applications to move out continuously while rendering it easier for corporate customers to build their own non-standard applications.

But no factor was more important from an investment standpoint than the unique revenue model with which the enterprise software industry was endowed by a benign providence: the "perpetual, non-exclusive license" to run purchased software, bundled with an annual maintenance fee typically priced at 20% of the license as compensation for the vendor ensuring that the software would continue to run as the operating environment evolved.

There is no definitive explanation for how the license model for acquiring software arose at the end of the 1960s in a world where IBM was more than simply a dominant competitor in the market for commercial computing: IBM was the "environment." IBM established the standards and defined the terms of engagement for the other players, whether niche computer companies or supportive service providers. IBM's grip on its customers was sustained through its own distinctive business model: it leased its computers so that incremental functionality—whether delivered as hardware or as software upgrades and add-ons—were mindlessly easy to sell and to purchase.[1] But IBM had come under anti-trust assault from the U.S. Department of Justice and, in 1969, its response was to unbundle software from its hardware leasing contracts, thereby opening up commercial space for an independent software industry. And over the following generation, the perpetual license model served as the financial engine for the growth of that new industry.

The importance of this extraordinary revenue model is simple to grasp: by purchasing an upfront license, the rich customer was providing the cheapest capital imaginable to the definitionally poor enterprise software start-up. Venture capital was needed for the initial cost of developing the software, including the cost of buying development hardware and software, but once there was a "minimal viable product" (to use a modern term) to sell, license payments funded the build out of the sales, marketing and customer support functions essential to turning a venture into a sustainable business. Moreover, the enterprise software industry was the first product-centric business not to require the "manufacturer" to hold inventory: losses on the revaluation of inventory were eliminated by construction and working capital needs were limited to financing receivables. With good execution, the nirvana of positive cash flow from operations could be reached with no more than $20–30 million of 1980–1995 dollars. And, from 1982 on to 2000, a welcoming IPO market was available to fund growth and acquisitions and to provide liquidity to the venture investors.

In enterprise software, especially at the application level, we were selling a chassis with a more-or-less working engine and some of the basic tools for driving, like a steering wheel and accelerator and brake pedals. The customer understood that additional components, such as a windshield or headlights, would be added in further releases down the road.

The perpetual, non-exclusive license model did raise complex and occasionally intensely litigated issues around revenue recognition. Standard procedure was to recognize all license-revenue upfront, along with the cash received. But what was typically being sold and bought was not a finished product, more of a work in process as additional functionality was added and bugs were removed, with the follow-on work paid for by the annual maintenance fees. I remember attempting to explain this to a newly recruited CFO who came from the auto industry. In enterprise software, especially at the application level, we were selling a chassis with a more-or-less working engine and some of the basic tools for driving, like a steering wheel and accelerator and brake pedals. The customer understood that additional components, such as a windshield or headlights, would be added in further releases down the road. The ability to convince the customer of the value to be realized when the purchase finally became a fully functioning vehicle was why enterprise software sales people were typically the highest paid employees in the company, distinguished from afar by their gold Rolex watches.

Given the degree of discretion—legitimate or illegitimate—associated with revenue recognition of the sale of software and related services, prudent investors have always been on notice to "follow the cash." And successive revisions of accounting standards to be applied to "software entities" have served to increase substantially requirements for the exercise of managerial judgment with respect to such factors as identification of performance obligations and the satisfaction thereof and realizable versus contractual transaction prices.[2] Consequently, actual movement in cash balances from period to period—not the accounting entity sanctified by the private equity industry as "EBITDA" (earnings before interest, taxes, depreciation and amortization)—was and remains the appropriate metric for evaluating financial performance relative to operating activities.

While alternative revenue models were available and occasionally deployed, such as term licenses, the perpetual license model dominated through the great Dotcom/Telecom/Internet Bubble of 1998–2000. The two enterprise software companies with which Warburg Pincus was most deeply and rewardingly engaged, BEA Systems and VERITAS Software, each demonstrated the ability to leverage that model into more than $1 billion in highly cash-generative revenues within barely five years. Since these companies were delivering crucial infrastructure to the newly emergent web architecture—the WebLogic application server and the VERITAS data management suite, respectively—their products had to work out of the box: they did and the rest is business and investment history. But the enterprise software industry that they exemplified is dead as target for entrepreneurial investment. The revenue model and the market environment for enterprise software have both been transfigured.

Actual movement in cash balances from period to period—not the accounting entity sanctified by the private equity industry as "EBITDA"—was and remains the appropriate metric for evaluating financial performance relative to operating activities.

WHAT IS NEW AND DIFFERENT?

Start with the revenue model. Salesforce.com was launched in 1999, just as the Bubble reached its apogee, with a novel approach: Software as a Service ("SaaS") transformed the customer's purchase decision from a capital investment to a recurring operating expense.

A product sale now generated cash flow to be collected in the future, whether based on a simple calculation of number of users licensed per period or transactions executed or some other metric of value realized over time versus anticipated value paid for upfront.

For the vendor it eliminated the high-risk, lumpy quarterly license challenge (and with it the temptation to cheat by holding the books open for the license sale that would make the quarter). But, while the SaaS model made it radically easier to sell software and to forecast reported revenues as contractual payments were made over time, it came with a cost. Salesforce.com was the first enterprise software company characterized by sound operating execution to consume more than $100 million of funding to reach positive cash flow. Now the poor start-up was in the role of financing the rich customer. Funding from launch to positive cash flow for a SaaS enterprise software company runs from that $100 million to twice as much or more, some five times the $20–30 million of risk equity once required to get a perpetual license enterprise software company to positive cash flow.

The SaaS revenue model has become canonical, but it is not the only contributor to the transfiguration of the enterprise software industry. Four other profound changes have transformed the world in which enterprise software firms operate:

The maturation of the open source software movement complemented by the adoption of "agile," iterative development methodologies, radically less risky than legacy linear "waterfall" methods but requiring a fundamentally different mindset on the part both of developers and managers;
The emergence of "the Cloud" as an abstracted computing environment for development and deployment;
The reduction of the responsibilities (and the budgets) of internal corporate IT departments largely to "keeping the lights on" and to protecting the corporate systems from cyber attack.
And, finally, the persistent decline in access to the public equity markets for all venture-backed firms and especially those in the IT sector.

The first two, together, mean that the absolute and, even more, the risk-adjusted cost of launching a new offering has been radically reduced. With software tools available for free and computing resources available for rent, the upfront cost of building enterprise-class software has declined by a decimal order of magnitude. At the same time, reliance on open source tools has raised the prestige of contributions to their continuing evolution, accelerating the functionality and scope of literally free software. And agile development of programs that have been broken down into discrete deliverables radically increases the likelihood that want is being coded will actually work and satisfy the target customer's expectations. Moreover, cloud-hosted business software requires minimum friction in deployment and lends itself to the delivery of rapid bug-fixes, performance improvements and incremental functionality without the technical and marketing risk s to customers and vendors alike of major, discrete releases delivered after intervals of a year or more. Above all, of benefit to suppliers and users alike, only one version of the software exists at any time.

Implications for Building an Enterprise Software Business

But the cost of building a business, complete with all of the resources needed to sell and market and service such offerings, has hardly declined at all. So a Darwinian explosion of "hopeful monsters" is followed by the intensely competitive process of market selection applied through a much longer ramp to positive cash flow. While those who do get to scale enjoy an extremely attractive cash flow profile, the time and money needed to reach scale can be daunting. The emergence of Amazon and Microsoft, with Google following, as increasingly trusted hosts for enterprise software has had another effect. Functionality that was traditionally sold to the enterprise customer is now delivered as a component of an increasingly comprehensive and progressively "thicker" bundle: "Infrastructure as a Service" becomes "Platform as a Service." The Cloud providers are pre-empting market space that was once served by independent infrastructure software companies.

The third factor reinforces the first two. For within the enterprise client there is no longer a capable IT customer charged with delivering application solutions to the business units of the enterprise, both those facing outwards like sales and those facing inwards like HR. BEA sold an application platform to the enterprise IT customer who in turn would deliver an array of functional systems to complement or extend the more or less packaged, standard enterprise applications. No more.

The evolution of the competitive environment is accelerating across industries with the "consumerization" of IT: "bring your own device" to work and with it the expectation that interactions both with and within the enterprise will be as seamless and transparent as with Facebook or Instagram. The enterprise "customer" is now fragmented into multiple customers: the disparate business units where the need and budget are to be found…but not the technical capacity to deploy or manage complex code. And the cost of building a channel to those distinctive and distributed customers is typically greater than to the centralized IT department, even as the cash flow generated from each successful sale has been spread out over time.

Cloud providers are pre-empting market space that was once served by independent infrastructure software companies.

In consequence, the role of enterprise IT is increasingly focused on after-the-fact integration both of access to the point solutions purchased by line of business customers and of the data generated and captured by those applications. And those challenges are amplified by the urgent need to secure both the distributed applications and data from leakage, let alone from deliberate attack. And so the key leaders in enterprise IT have become the Chief Data Architect and the Chief Security Architect. The days when the Chief Information Officer was charged with designing and implementing a centralized IT architecture for the enterprise are gone.

Given that SaaS-driven start-ups need five times as much risk equity to reach positive cash flow, the post-Bubble decline in the U.S. IPO market has differentially affected enterprise software ventures. When the SaaS model was new and its business appeal was apparent, a few new companies managed to execute IPOs early and fund their growth to positive cash flow with cheap public equity. But the number of VC-backed IPOs has declined from an average of 30 per quarter during the two decades prior to 2000 to about 10–20 per quarter since 2000.

This actually understates the impact on enterprise software as there has been a striking relative increase in BioTech/Medical IPOs, reflecting their unique attribute: in a 3rd party payer system, the market is assured for an innovation that receives FDA approval. And the absence of market risk has been amplified by the increased acquisitiveness of Big Pharma in response to the evident decline in the productivity of internal research. Finally, given the necessarily limited number of actual users of enterprise software compared with the billions of potential users for consumer web services, the Unicorn Bubble scarcely touched the former category of start-ups while it expanded: seemingly limitless amounts of private funding at inflated valuations certainly cannot be relied on as the Unicorn Bubble deflates.

OPPORTUNITIES AND RISKS

So what is the enterprise software entrepreneur and her venture backer to do? It is not as if there are no opportunities left for radical innovation. Application solutions are of two sorts: "horizontal" solutions that automate business functions across a diverse set of industries (like Salesforce.com itself) and "vertical" solutions that automate deep layers of functionality for particular target industries (like "know your customer/anti-money-laundering" software for the financial industry). Drivers of innovation in each segment are available, for example: (1) technologies to extract meaningful information from Big Data and to integrate that analytical information with transactional systems in real-time and (2) technologies to integrate mobile devices with enterprise transaction and analytic systems fully and securely. Both sets of opportunities require technically challenging extensions of the infrastructure while enabling new categories of applications.

The first step is to identify clearly the dimensions of risk that face every start-up venture, while taking account of the transformed context in which each must be addressed today:

Technology Risk: "When I plug it in, will it light up?"
Market Risk: "Who will pay to buy it if it does work?"
Financing Risk: "Will the capital be there to fund the venture to positive cash flow?"
Business Risk: "Will the team manage the transition from start-up to sustainable business, especially given the challenge of building an effective channel to the market?"

Given Open Source and the Cloud, the cost to overcome Technology Risk—or to fail to do so—is far lower, as noted. Given SaaS, the magnitude of Financing Risk, also as noted, has risen greatly. Business Risk remains, as always, the least certain since it remains so dependent on the "soft" skills of entrepreneurs ("there is no business so good that it cannot be ruined by incompetent management"). But it is along the dimension of Market Risk that the most complex process of change has taken place.

The disappearance of the internal enterprise IT Department as the customer for software means, in the first instance, that attempting to sell a new general purpose "platform" to the enterprise—emulating the enormous success of Oracle in database or BEA in app server—is quixotic. There is nobody home when the salesman calls. Instead, the technology must be fashioned into a "solution" relevant for and appealing to the distinct business unit customer. Now the problem is that there are too many homes on whose doors to knock.

Thus, a commercially viable technology platform today must be redefined as one that is built upon an open source stack, caters to very specific use cases to begin with, and is extensible with configuration (not customization) to a broader set of applications. These platforms start with a vertical domain application, necessarily with a constrained initial total addressable market (TAM), are relevant to a specific set of business users, are sold as domain-specific solutions, but can be extended to other domains (giving them access to increased TAM over time) by adding domain and business knowledge to the platform team and configuring new use cases for the technology team. In effect, the "platform" will enter through the back door, pulled along by the demonstrated value of the solutions that run on it.

Examples of some such platform areas, in no order of priority, are:

Intelligent crowdsourcing (starts with next generation BPO use cases)
Unified commerce (starts with intelligent retail use cases)
Internet-of-Things (starts with field monitoring and remediation use cases)
Social engagement and learning platforms (starts with next generation CRM use cases)
Field services and distributed workforce management (starts with field action use cases)
Machine learning and analytics (starts with time sensitive decision use cases)
AI and NLP platforms (starts with experience, fraud, and risk use cases)

Finding and selling the business buyer means bringing to bear skills in marketing and product management sufficient to understand those distinct business needs and to qualify which ones are appropriate targets, as well as the technical skills to transform a general purpose platform into a targeted solution. The alternative is to package the core technology into service offerings that are sold and delivered as projects: potentially very high value ones.[3] But a project-based business is inherently incapable of the rate of growth and the profitability of a product-based one.

The arithmetic logic is inescapable: the greater the degree of focus on the IT needs of a particular type of business user, the smaller the addressable market. Thus, the most critical strategic task for management and the board is the mapping of functional capabilities to market needs such that the scope of the solutions delivered are broad enough to offer attractive investment returns on the capital required to reach scale operations and positive cash flow. Contrariwise, selling to the technologically unsophisticated business buyer of software is compromised by that buyer's need for help in sorting through the competing claims of vendors. Gartner Group and its competitors historically provided such guidance to the IT Department, but the fragmentation of business buyers and solutions requires a different sort of granular analysis.

The most critical strategic task for management and the board is the mapping of functional capabilities to market needs such that the scope of the solutions delivered are broad enough to offer attractive investment returns on the capital required to reach scale operations and positive cash flow.

The sustainable enterprise software businesses that successfully meet today's challenges will be characterized by extraordinary discipline both in execution and in focus, able to answer the perennial questions for every technology entrepreneur: tell me again whose problem you are solving and are there enough of them to make the exercise worthwhile? The new environment offers some compensation by enabling access to the market segment ill-served by the old model: the small/medium enterprise ("SME") market. The cost of the direct channel for sales and support was previously just too great. But with the web as a marketing and sales channel for cloud-hosted horizontal solutions, the cost of delivery and support now matches the revenue to be realized…even while the path to sufficient scale to generate positive cash flow remains long.

An existence proof of the ability to negotiate this new landscape is represented by Blue Yonder: a German software company delivering predictive analytic solutions to retail enterprises (full disclosure: Warburg Pincus is Blue Yonder's largest shareholder). Blue Yonder has fashioned its set of machine learning technologies—potentially adaptable to almost any exercise in probabilistic prediction—into targeted solutions for retailers facing Amazon's assault: demand forecasting, pricing and replenishment. By successfully integrating analytics with transaction processing at very high volume, Blue Yonder has defined a market for its solutions fully worth addressing. And it has the opportunity, based on demonstrating the commercial value of these solutions to bring into the enterprise the platform technologies that distinctively enable those solutions.

In sum, the investments required of a new-style platform player will exceed, perhaps substantially, those for a targeted, one-off point solution venture. The new player must in parallel: identify specific, unserved or inadequately served application needs; develop the deep understanding of successive targeted domains; fashion its technology into a series of compelling solutions; and support the business customer with whatever services are needed to assure successful deployment and operations.

CONCLUSION: DO WE SELL NOW?

The general conclusion to be drawn from this assessment is, in the end, quite straightforward. Given the much extended path to self-sustaining positive cash flow from operations, the responsible entrepreneur and VC are charged with asking themselves the same question at each step along the way:

The technology does "light up"—do we sell now?
We have three credible customers who will testify that they have bought our offering and will buy more—do we sell now?
We have access to another round of capital but it will be dilutive and Business Risk looms—do we sell now?

For venture investors in this domain, the default definition of success has come to be represented by a timely trade sale rather than by a successful IPO and subsequent exit by distribution or sale of shares into a vibrant public market. The Cloud companies in particular—Amazon, Google, Microsoft—are showing themselves to be aggressive acquirers of technology for building out and enriching the platforms they offer, with some demonstrable ability not to smother the start-ups they absorb. At the frontier of web-based services for individuals and businesses, respectively, Facebook and Salesforce are also very much in the game. And so are the two dominant ERP vendors—Oracle and SAP—especially in pursuit of acquisitions that can help them plausibly claim to be delivering cloud-based solutions rather than the hard to deploy and manage on-premise applications on which they were built.

For venture investors, funding distributed research and development for acquirers may not present the heroic profile of yore. But it is a valuable—even critical—contribution to the innovation economy, especially when the pace of change exceeds the inventive capacity of established enterprises. In fact, the systemic constraint has shifted: from the problematic ability of start-ups to reach critical mass and sustainable cash flow as independent businesses to the ability of large, established companies to exploit innovation from the outside.

That innovative ventures will continue to launch is certain. Those ventures will continue to push back the frontier of computing capabilities, accelerating the digitalization of the commercial and financial world. The enterprise software model that had a 3o year run to 2000 may have died, that is to say, but a new one has emerged, transfigured into alignment with today's transformed environment.

William H. Janeway, Warburg Pincus and Cambridge University

Raj Kushwaha, Warburg Pincus

October 23, 2016

Notes

[1] In the late 1970s, IBM chose to induce its customers to switch from leasing to outright purchase, artificially sustaining double-digit revenue growth for a time at the expense of reducing its market control.

[2] EY, “The New Revenue Recognition Standard—Software and Cloud Services,” January 2015, available at http://www.ey.com/Publication/vwLUAssets/Applying_IFRS_in_Software_and_Cloud_Services:_The_new_revenue_recognition_standard_-_software_and_cloud_services/$File/Applying%20IFRS-Tech(Software)-Rev-Jan2015.pdf

[3] Palantir’s cyber security business is an example.