Under the Buzz NUGGETS

Commentary on Business Strategy for Tech Company Executives & Professionals

Vol. 12, Number 2 – May 3, 2011

Under the Buzz is an email "viewsletter" published by Philip Lay, managing director at TCG Advisors, a Silicon Valley-based firm that helps executive teams in tech companies to deal with complex strategic, organizational, and operational challenges. Now in its twelfth year, this journal is published periodically and delivered free to subscribers via email on an opt-in basis. It is also posted on TCG Advisors' website, http://www.tcg-advisors.com/Library/utb/utb.htm, where back issues are also available.

What to Say When a Competitor Screws up and Your Category is Under Fire

Below is a best-practice example of how to do damage control on behalf of your (emerging) category when a competitor experiences a serious problem – in this case, a severe outage that brought down Amazon's cloud services for two days. Oftentimes executives find it hard to resist taking advantage of a competitor's travails to sing their own praises, not realizing that they are doing greater harm to their product/service category. In this case, Lew Moorman of Rackspace took the high road in exemplary fashion, and in the process put the Amazon outage in clear perspective:

"Thursday's Amazon (cloud services) interruption was the computing equivalent of an airplane crash. It is a major episode with widespread damage. But airline travel is still safer than traveling in a car – and cloud computing is safer than data centers run by individual companies. Every day, inside companies all over the world, there are technology outages. Each episode may be smaller, but they add up to far more lost time, money and business. … We all have an interest in Amazon handling this situation well."

- Lew Moorman, Chief Strategy Officer, Rackspace Hosting – NYT Saturday, April 23, 2011

Main Article: **Pursuing Platform Power**

Do you often experience platform envy? Go on, admit it, you're dying for your company to be like Apple, owner of at least five of the most powerful platforms in technology today – the Mac OS X, the iPod digital media player, the iTunes digital media service, the iPhone smart-phone/computer, and the iPad tablet computer. If not Apple, then Google, with its still-dominant search engine; or Facebook, with its seemingly ubiquitous social networking portal; Amazon,

with its best-in-class online shopping system; eBay, with its uniquely successful online marketplace now being augmented by Paypal, its fast-growing payments platform, and so on. In the enterprise computing world, we've seen the IBM mainframe – one of the original dominant IT platforms - the PC – one of the very first open platforms, the Unix platform that Sun and HP modified and alternately dominated, Oracle's relational database management system, SAP's integrated suite of ERP applications, VMware's virtualization platform, Salesforce.com and its emerging Force.com custom application development platforms alongside its AppExchange enterprise application marketplace.

Platform, platform, platform, it seems that this is all we hear about these days. And we haven't even mentioned the venerable Windows OS and Office suite of productivity applications that Microsoft drove to prominence and massive profitability starting twenty-five or so years ago. The most successful platforms in tech all become household names: others include Cisco, Yahoo! (portal), Adobe (Acrobat PDF document format), Autodesk (DWG design format), Nokia & Motorola (mobile phones), and so on.

Before we go too far along with this line of thinking, let's make sure that we agree on the essential difference between products and platforms. After all, are most if not all of the offerings listed above not product offerings in their own right? And, if so, what makes them *both* products and platforms? Platforms generally start life as free-standing offerings, in the form of a product or service. Later on, once they have achieved successful adoption as products aimed at solving specific problems, they may become reference platforms. For this to occur, the following three attributes are vital pre-requisites:

1. <u>They achieve leading to dominant share in the marketplace</u>, thus becoming the *reference* competitor for all other players to beat. To make this very clear, they not only have to cross the chasm in terms of adoption, but they have to experience some kind of "tornado" as the acknowledged leader, or as a credible candidate to become the definitive leader once the wars for market leadership have been won/lost.
2. <u>They provide a platform (literally) for third party companies to develop and market complementary products and/or services</u> – and thus they make a market for others, which in turn creates a self-reinforcing ecosystem around the platform, thus strengthening the market power of the platform owner or sponsor.
3. <u>They enjoy network effects</u>, which can be both direct (for example, an interface standard) and indirect (for example, adhesion by an overwhelming number of application developers, service providers, and others). [*]

Undoubtedly these three attributes represent a high bar; how about platforms that don't achieve dominant adoption as *the* uncontested winner in their category - can they not become relevant and generate significant business for their sponsors? If you look at a very current platform war, the one between Apple and Google for setting the smart-phone standard, they each have achieved power in slightly different ways. Apple forever changed the dynamics between mobile-phone manufacturers and carriers by virtue of the irresistible appeal of the device to consumers combined with the market power of the AppStore; Google followed in Apple's footsteps and at this year's Mobile Congress in Barcelona was the star of the show, eclipsing even further the carriers.

But whereas Apple quite deliberately remains a proprietary platform, Google's appeal has been its openness. One is a perfect example of a proprietary platform, the other a highly successful example of an open platform. What these two types of platforms have in common is that in order for other companies – third-party software developers or services providers, for example – to leverage them, the platform's sponsor/owner must deliver a reference architecture that provides APIs (Application Programming Interfaces) or the equivalent that will enable partners to hook into the (closed) technology. Among open platforms, the most extreme version to date has been open-source platforms, whereby large portions of the entire technology "stack" in the relevant architecture are opened up to permit crowd-sourced development rather than development only by named specialists.

One recipe that tends not to work is for startups and even more established companies to come right out of the chute with what they claim to be a "platform". Adoption really does need to precede platform success, for the simple reason that without clear acceptance by customers no third party can have any basis for risking an investment in your (unproven, possibly irrelevant) platform offering. As self-evident as this statement might be to some, there are thousands of tech startups that launch their first product as a platform upon which they hope other companies will build products and services. The only near-certainty they face is that they will be thrust into the Chasm sooner than they realize, and will probably remain there unless and until they realize that they must get out and sell their platform technology as part of an application that solves a business problem for at least one type of organization.

More on Open Platforms

Truly open platforms are a relatively new phenomenon. The first two open hardware platforms were probably the IBM-designed PC running DOS (launched in 1981), and "servers" running the Unix OS that AT&T had developed in the late seventies. Unix on the then-new client/server platform was quickly co-opted by IBM, HP, Sun, and others who developed their "proprietized" versions of the Unix OS. The first breakthrough for the Unix client/server OS came when (a) the Oracle and Sybase relational database management systems became prevalent, along with the Powersoft development environment for building applications. But the critical breakthrough that proved to corporations and government agencies that the Unix platform was for real came when SAP and other ERP companies demonstrated that their software could operate in a mission-critical environment.

Unix have the benefit of being semi-open and it also preceded the arrival of more truly open platforms, such as the Internet, its IP protocol and HTML language (1993), Sun's Java programming language (1996), Linux open-source OS (circa 2002), and more recently the Google-led Android alliance (2007) mentioned above, the Rackspace-led OpenStack initiative for cloud infrastructure (2010) and even more recently VMware's announcement of the open-source Cloud Foundry (2011).

Platforms often spring up as movements to provide customers and the industry with an alternative to a dominant player. This is the case of Linux (against Unix, .Net and Windows), Android (against Apple), and the two cloud open-source initiatives (against the looming presence of Amazon, Google, and Microsoft). Specifically in VMware's case, its initiative appears to be an attempt to not be made irrelevant by the public cloud movement.

But more than anything else, what the sponsors of these open-platform initiatives must avoid at all costs is the urge to co-opt the movement they have started, once they have experienced acceptance. Right now, Google is being challenged on its decision to restrict the presence of competing search engines and browsers on Android phones. This may not only get the company in trouble, but curtail or limit the growth of Android as the different players in the ecosystem evaluate changes in their go-forward plans.

Best Practices for Platform Adoption

Overall, we've seen that, once a company has gained leading acceptance for its product offering, the Best Practices for gaining platform acceptance and leadership – added to the self-reinforcing network effects that this can generate - include these factors:

Strategic intent:

- Enlightened self-interest combined with deferred gratification: *"Make a market for others, before making a market for ourselves."*

Power positions to play from:

- Established category, company, and market power achieved by the product that is becoming a platform

Go-to-market strategy:

- Learn to deliver your own whole product/offer while simultaneously contributing to whole products designed and delivered by selected partners

Key programs/deliverables to motivate development of the customer and partner ecosystem:

- Thought leadership in the form of a compelling vision, strategy, and roadmap
- Coherent reference architecture with API plug-ins
- Deep collaboration between development /field teams and key-partner teams
- Whole product alliance management and resource commitment
- SDKs (Software Development Kits)
- Cooperative marketing programs
- Sales and technical training
- Technical certification program
- Consultative professional services approach
- Web services / open-source domain expertise
- Commitment to social media / community marketing and ecohubs

Indeed, this is a comprehensive list. The good thing is that your first platform marketing initiative may not require that you execute flawlessly on every one of these components and deliverables. But if you're serious about pursuing increased platform power, you must execute the most important ones.

[*] Michael A. Cusumano, professor at Sloan Mgmt School of Management, author of "Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World" (OUP, 2010), has written about technology platforms in prior publications and books, and some of his ideas have been a reference for this article.

This article was authored by Philip Lay, managing director at TCG Advisors.

Disclosure: From time to time, the author and/or his firm may hold investments in, or provide advisory services to, companies cited in this journal.