



Eclipse, Open Source, Wall Street and Competition: Big Drama, Big Money

Brent Williams
Managing Director, Equity Research
The Benchmark Co., LLC
bwilliams@benchmarkcap.com
brent_williams@hotmail.com

Big SEC-Mandated Legal Disclaimer

- Don't take my comments today as investment advice
 - ◆ Free investment advice is rarely worth more than you pay for it, anyway.
- Beware conflicts of interest
 - ◆ Benchmark Co. may seek to perform investment banking or other services with the companies I mention in this presentation.
 - ◆ Benchmark Co. may make markets in shares of some companies mentioned in this presentation.
- One spot of good news, however
 - ◆ I don't own shares of any of the companies we may discuss today

My Biography, Much of Which is True

- 10 years' experience as Wall Street stock analyst
 - ◆ I cover infrastructure software companies including operating system, database, middleware, development tools.
 - ◆ Have written research on Red Hat longer than anyone on the Street.
 - ◆ Have presented at EclipseCon last three years
- Almost 20 years' experience in software business
 - ◆ Unix/C engineer, working on Ingres database in early days of database business (1986-1990).
 - ◆ Sales and marketing experience in database and application server industries.
 - ◆ Analyst at IDC, in charge of covering Microsoft when Windows NT rolled out.
 - ◆ Analyst at Gartner, in charge of covering development tools



2007 in review: Big Splashy News Headlines

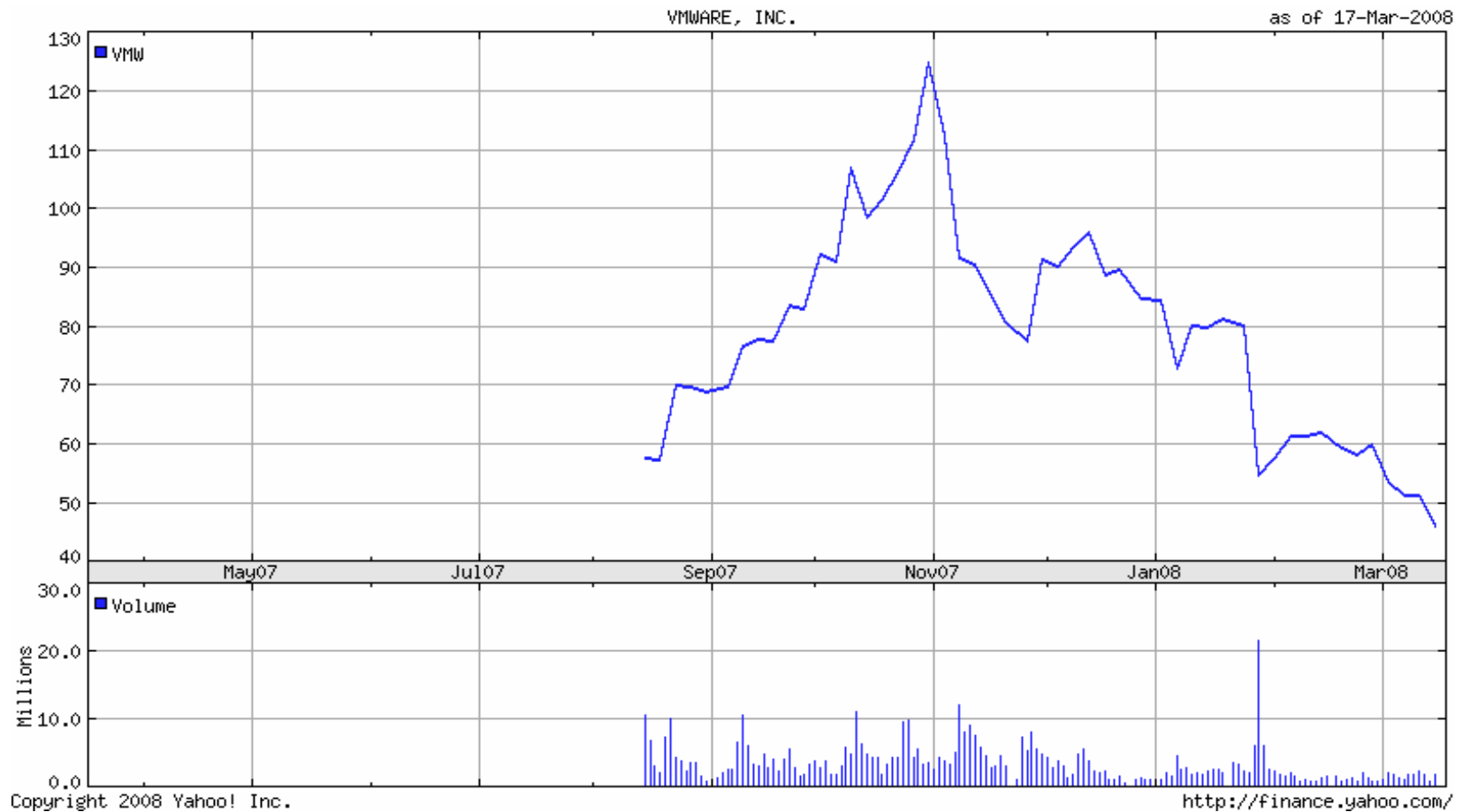
Recession talk

- Some people are positive
 - ◆ Federal Reserve: “We’re not in a recession and we’re cutting rates just to prove it!”
 - ◆ Oracle: “we’re recession proof”
- Some not so positive
 - ◆ Many industry analysts: IT spending growth rate in 2008 will be down
 - ◆ Gartner: Already telling CIO’s to prepare drastically reduced IT budgets
- Some downright dire (but also credible)
 - ◆ Roubini Global Economics: \$600 billion more bank write-downs to come; \$2,700 billion total writedowns; “worst US financial crisis since the Great Depression.” (17-Mar-2008)

Industry consolidation continues

- Buying for fundamentally financial reasons.
Poster child: Oracle→BEA
 - ◆ 4 middleware architectures/stories to meld: Fusion, AIA, BEA WebLogic, BEA AquaLogic.
 - ◆ Customers increasingly restive about return on maintenance spend. (5 years in, where's the beef?)
 - ◆ Middleware technology changes starting up again – wrong time to buy for financial returns
 - ◆ Market share ≠ market dominance or pricing power.
- Defensive Buys: “Don’t get left out!”
 - ◆ Consolidation of 3 diversified BI vendors in 2007
- “Strategic acquisitions” (a.k.a., desperation)
 - ◆ MSFT→Yahoo

VMWare: Hot IPO, Hot Technology



VMWare: Hot IPO, Hot Technology

- “Hottest IPO since Google...” at least for a while
- Shows that investors are willing to reward vendors of game-changing infrastructure software...at least for a while
- True meaning of virtualization?
 - ◆ More flexible platform
 - ◆ Respects legacy stuff
 - ◆ Great IT benefits
 - ◆ ... though end users don't “feel” any differently (that's all foreshadowing for Part 3 of this talk!)

Big proprietary vendors buying open source vendors: “Powdered Rhino’s Horn” or something more?

- Oracle-Sleepycat (February 2006)
 - ◆ Gee, that worked well...
- Citrix-XenSource: catch up with VMWare
 - ◆ Citrix needs velocity, squeezed between VMW and MSFT
 - ◆ May forget open source component of Xen
- Yahoo-Zimbra
 - ◆ Saved from pre-IPO poverty just in time to be hit by the Microsoft bus...
- Sun-MySQL
 - ◆ Driver for ZFS and other hardware-independent storage?
 - ◆ Jury still out on this one
- Nokia-Trolltech
 - ◆ Maybe it’s just because they’re both Scandinavian

Mega-Vendors' "Bambi vs Godzilla" Open Source Moves, A Year Later

- Oracle "Cloning" of Red Hat Linux still stubbornly refusing to generate revenue despite increasing lack of interest on Oracle's part
 - ◆ Oracle's Xen cloning effort disappeared even faster than the Red Hat attack
- Shows value of branding and customer loyalty on something that's supposedly a commodity
- Message to mega-vendors: if you don't contribute to a project, you don't sell a lot of it

Patent Wars

- Microsoft: “Linux infringes 235 of our patents!”
 - ◆ Linux: “Show me!”
 - ◆ Effect on open source momentum?
- Large-scale backfire against patent-driven software business model may be imminent
 - ◆ KSR-Teleflex case weakens patent trolls
 - ◆ EU still against software patents
 - ◆ Congress taking up patent reform issues
 - ◆ Open source methods applied against software patents: scale that no law firm can match

File Format Wars

- Exemplified by ODF-OOXML battles
 - ◆ Sleaziness of Microsoft behavior in ISO vote-garnering is highly visible
 - ◆ Would be nice if OOXML were defeated, but not necessary
- Unintended Consequence of OOXML Battle:
 - ◆ Customers become aware of value of file formats in an open systems strategy (we forgot for a long time)
 - ◆ Potential long-term boost to demand for XML and broadly standardized file formats (regardless of whether OOXML wins or not)

SaaS becomes a “real” platform

- Now slices value in both dimensions: horizontal and vertical
- Horizontal: Outsourcing infrastructure layers separately
 - ◆ Examples: Amazon S3/EC2, Google GFS, Bigtable
- Vertical: outsource standalone apps (CRM), modules (CNQR), transactions (ADP)
 - ◆ Salesforce.com becomes an application development platform

**All of These Developments might actually be
“Full of Sound and Fury, Signifying Nothing.”**

If so, what's the real change driver?

Real change driver is several subtle shifts

- New creativity in programming languages (again)
 - ◆ Typically happens after periods of stagnation in languages.
 - ◆ Ruby, PHP, Groovy, domain-specific languages, ...
- Storage software/hardware getting more interesting
 - ◆ No more “valuable bits” versus “second-class bits”
(database-like reliability comes to all types of data and alters value equation for a key part of the stack again)
- Database business fracturing and opportunities are increasing
 - ◆ Column stores
 - ◆ Stream databases
 - ◆ Content databases
 - ◆ Scalability issues in data warehouses

Sum of these changes implies brewing platform war

- Architectural assumptions holding together existing vertically integrated stacks are changing
 - ◆ Implying the value of integrating those stacks in that particular way may be eroding
- Existing platforms are starting to show their age
- Customer requirements are evolving, particularly in need for flexibility

... seems like we're overdue for one!

Key Indicators That Something New Is Afoot

- New database vendors (not just MySQL) starting to make inroads
- Google and SaaS starting to put existing players on the defensive
- Facebook, MySpace, LinkedIn as platforms
 - ◆ They've got real APIs and *everything*...
- Continuing interest in Web 2.0
- Initial hysteria about “cloud computing” and “mesh”
 - ◆ “My PowerPoint architectural vaporware can beat up your PowerPoint architectural vaporware”
 - ◆ ...people are casting about to see if there already *is* a new platform that they can latch on to

How to Think About a Potential New Platform

- Past platform wars
- What's wrong with current mega-platforms?
- What is going to disrupt existing platforms?
- So what's the problem?
 - ◆ Platforms increasingly unresponsive to customer needs.
 - ◆ Platforms strangle themselves with complexity and resistance to change.
- One paradigm for connecting layers of a stack

When Platform Shifts Happen: High Level Theory

- Spread between fastest and slowest devices widens 10x-100x
 - ◆ 1970s-1980s minicomputer wave, client/server Windows era
- New programming models
 - ◆ Structured COBOL, DLLs, OOP, CASE, SOA, AOP, DSLs, ad infinitum
- New user interface technologies: Block mode terminal → character mode terminal, GUI, Web, SOA (machine-to-machine “UI”), Mobile
- New hardware technologies
 - ◆ Storage arrays increase reliability of *all* data – narrowing distinction between highly protected data and “ordinary” data
 - ◆ Solid-state disks changing game for transaction processing applications – transaction commit window becomes delightfully small, reducing value of specialized transaction engines
 - ◆ Creates opening for non-relational mission-critical data, and radically different architectures between OLTP and other systems
- When one-size-fits-all platform no longer fits all, there’s an opening to exploit

Deep Dark Past of Platform Archaeology

- 1st-3rd Generation Proprietary Platforms (1965 - ~1995)
 - ◆ Mainframe, mini, Windows PC
 - ◆ Extreme architectural & vendor lock-in
 - ◆ Lower complexity enabled fantasy of “one vendor could do it all.”
 - ◆ Winners harnessed broadest addressable markets
 - ◆ Cost of lock-in could be glossed over because IT as % of revenue low, even in IT-intensive industries
- Galactic Architectures (a comical detour, 1987-1995)
 - ◆ Everything to everyone, solving all problems
 - ◆ IBM SAA, Digital AAS, Apple VITAL, ACE Consortium, Sun ONE, etc.
 - ◆ Some (SAA) had high vendor lock-in; some (DCE) only platform lock-in
 - ◆ Some had neither (Apple VITAL) but also had no *raison d’être* or business model
 - ◆ Often motivated by a *vendor* problem (defending against a new, cheaper competitor)
 - ◆ Collapsed under technology weight or utter lack of customer benefit

Current Dominant Platforms (.NET, Java)

- Current platforms succeeded because they learned from the past
 - ◆ Java: (relatively) open licensing model for multi-vendor support
 - ◆ .NET: multiple language support, focus on ease-of-development.
- But both have increasingly apparent holes
 - ◆ One data storage model and one transaction model at the center
 - ◆ One programming model (Java, .NET Framework)
 - ◆ One user interface metaphor (though fragmenting somewhat in the case of Java)
 - ◆ .NET tied to single (unappealing, high-cost) vendor for most key components
 - ◆ Both now seem to be in “defensive” mode, signaling maturity and slowing platform evolution
- By the way, they’re still vertically integrated just like past generations

What Next-Generation Platform Might Look Like

- Lowest layer more flexible: multiple data storage architectures and transaction models – drives everything else to be more flexible
- Wider range of client devices must be accommodated (RCP, the next generation)
- Machine-to-machine interfaces (Web services/SOA)
- Exploiting continuing re-slicing of value equation
 - ◆ Industry standards allow “slicing” of value equation in multiple dimensions.
 - ◆ Thus, there is lower perceived value for deep vertical layer-to-layer integration in customers’ minds
- Multiple architectures for integration between platform layers
 - ◆ Enables support for legacy with room for future new technologies
 - ◆ So it’s not really a “stack” at all
 - ◆ Redmonk calls this the “Stackless Stack”

Database Is The Key Architectural Component That Drives Flexibility

- Holes in Relational Model are appearing
 - ◆ XML/XQuery
 - ◆ Search
 - ◆ Stream-oriented data (not just a niche data model)
 - ◆ Moribund new feature sets from large existing vendors
- Synchronization and replication of data becoming more reliable and faster
 - ◆ Server-server remains important
 - ◆ Server-device is a new thing: Google Gears, JSR170, publishing data on RSS feeds.
 - ◆ But this is relational ⇔ non-relational replication
- Storage hardware gives much of the reliability of databases at lower cost
 - ◆ No longer is there a small category of “most valuable bits” that justify high cost software
- New transaction models and architectures to store that data reliably (“death to 2PC”) are emerging
- Mixed relational/non-relational designs in applications
 - ◆ Blending of content management and classical transaction infrastructures

This ultimately drives change and flexibility through the entire stack!

Who designs the new platform?

- Established large vendors?
- Small innovative startups?
- The end users themselves?
- Open source communities?

Mega-vendors Aren't (Really) Scalable

- The justification for bigness
 - ◆ “God is on the side of the largest army” (Napoleon, the day before Waterloo)
- The reality (in engineering): “Adding more people to a late project makes it later”
 - ◆ Frederick Brooks, The Mythical Man Month (1973)
- Lesson: Size \neq Scalability
 - ◆ Scalability is good to have in a market where technology is evolving; size can be a disadvantage when innovation starts

Small Startups are Too Broke

- VC's currently excessively enamored of “market of everything” (Facebook, Google, etc.)
- Will take a *lot* of money to build next-generation platform, so it had better sell into “market of everything”
- Will take a lot of time

... likely too expensive/slow for individual vendor to build next-generation platform

End users have tried to drive platform evolution before

- Avalanche Consortium
 - ◆ Large end-users sharing code amongst themselves (“software flea market”)
 - ◆ No commitment to joint R&D (“as is, where-is”) without requirement to contribute evolved code back to original contributors
 - ◆ No vendor involvement, meaning no R&D funding to turn cast-off customer-specific code into a “platform” (or even into an application)
- DrKW middleware consortium – OpenAdaptor
 - ◆ Died because vendors didn’t see what’s in it for them (originally a Tibco replacement)
 - ◆ Niche market too small for vendors
 - ◆ Many prospective customers already had legacy solutions
 - ◆ Especially hard for vendors when it sort-of-competed with proprietary vendors’ middleware stacks
 - ◆ Did all the right things, but have about 4 customers after 10 years

Can Open Source Deliver What End Users Need in an Enterprise Platform?

- So far: Open source increases odds of bug fixes and platform reliability because of multi-vendor environment
 - ◆ Vendor you're paying for support is first line of defense
 - ◆ Free community assistance from the mailing list
 - ◆ Can always find and pay third party to fix a bug
 - ◆ Open source licensing model can drive bug fixes back into main line code reducing repeated support costs on same bug
 - But that's where it stops today. Open source still weak at:
 - ◆ Customer involvement in all phases of creating and maintaining new platform (i.e., architecture to release)
 - ◆ Customer involvement in ongoing feature evolution
- ... Open source is not *inevitably* the provider of the next platform of choice – something more is needed...**

Open Source + Organization Is The Key

- Have to have governance process that works in wide variety of circumstances:
 - ◆ Competing overlapping situations (competitors collaborating in an open source project)
 - ◆ Disjoint independent situations (mobile device makers, database vendors)

...There's one organization in open source that has mastered this already...

How Eclipse stacks up (so to speak) technically

- Leverages existing platforms rather than rejecting them (e.g., extends Java platform rather than introducing something else)
- Architectural room for new tactical technologies (PHP, other languages)
- Beginning of platform-like components (RCP, OSGI, BIRT)
 - ◆ Framework is flexible enough to support addition of more platform components over time because it doesn't make core assumptions (database, etc).
- Not nailed down to a particular vertical layering architecture

How Eclipse stacks up economically

- Governance model for shared R&D between competitors is proven. “Manage the chaos” rather than “pick the ultimate winners” reduces politics and maximizes code delivery
 - ◆ Should be scalable to include competing customers as well as customers who couldn’t care less about each other
- Spread R&D model allows good returns when platform focused on multiple smaller addressable markets
- Architecture enables leverage of existing technology to recombine for multiple smaller addressable markets
- Some successes already under way with vertical market consortia (e.g., Autosar)

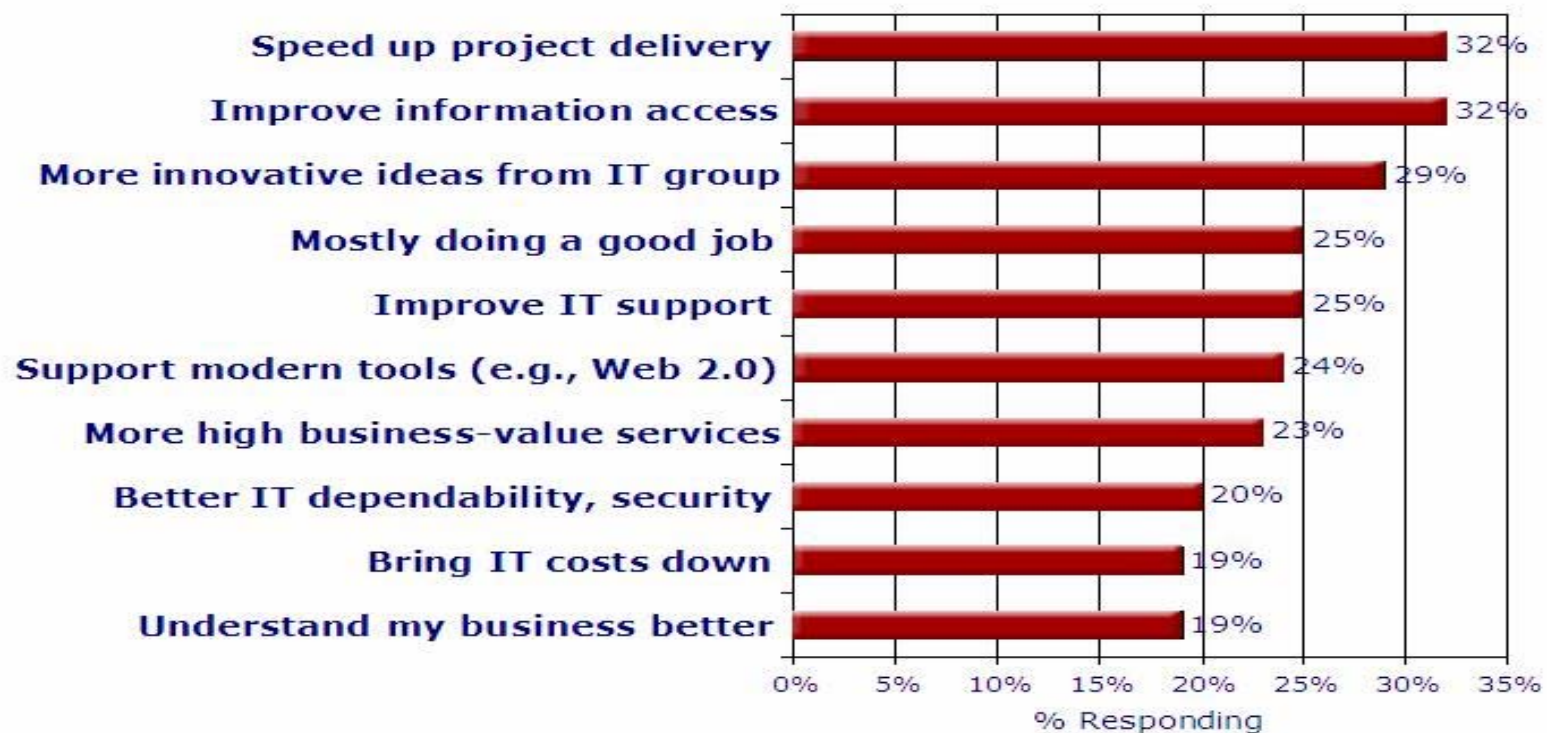
...but this is *still* not enough!



A New Twist to Open Source *Might* Work: End Users *Plus* Vendors Working Together

What Business Management Wants from IT (per IDC)

Q: Which of the following are the most important messages you would like to impart to your CIO/senior IT management?



Source: IDC QuickLook Survey, IDC's Enterprise Panel, January, 2008; LOB execs only (n=101); multiple responses allowed

Minimum IT *Needs* From Platform Vendors

(lowest to highest difficulty/importance/immediacy)

- Bug fixes for current operational problems
- Support for new underlying platform components (i.e., certification on new version of database)
- Specific new features to aid in solving specific issues
- General march of new features to broaden suite of problems that can be attacked with the platform

Traditional ways for customers to get what they need

Call tech support and yell

- Kind words
- No action

File a bug report via Web

- No kind words. It *is* a web site.
- *Usually*: no action
- *Sometimes*: bug fix in next release, at vendors' convenience

Call product manager and yell

- Kind words
- No action

Pay for user conference trip to talk directly to engineers

- Cheap canvas or nylon tote bag
- Convention center food
- (Geeky) kind words
- No action

Call salesman and threaten to cancel order if bug not fixed

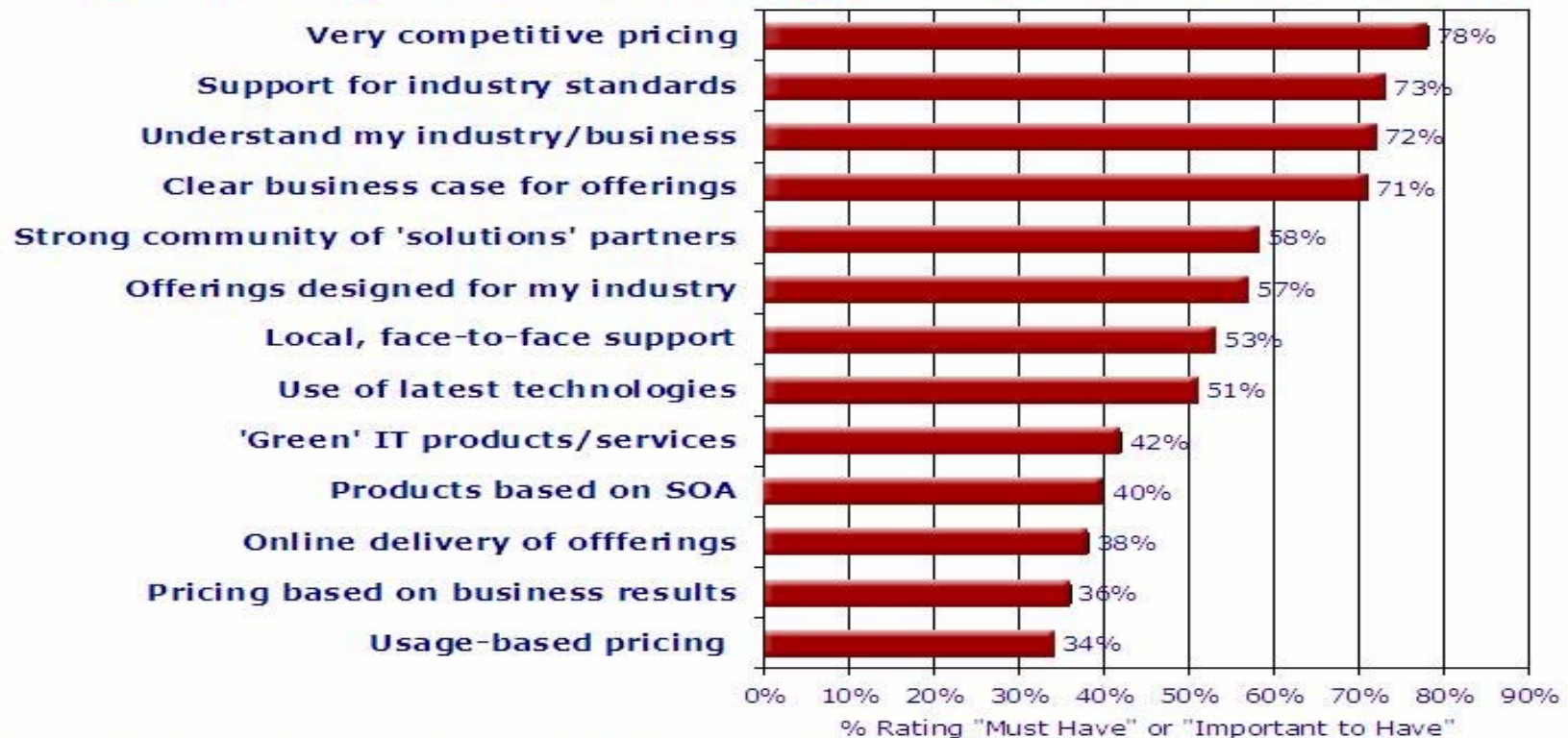
- Expensive lunch
- (Slick) kind words
- No action

Get onto "customer advisory board" and yell at CEO (only if you're a big enough manager at a big enough customer)

- Golf with CEO
- Expensive leather tote bag
- Free lunches *and* dinners
- Kind words
- No action

What IT Organizations *Want* from Vendors (per IDC)

Q: How important is it for your IT vendors to offer the following, when considering doing business with them?



Source: IDC QuickLook Survey, IDC's Enterprise Panel, January, 2008; LOB and IT execs (n=245)

Managed Customer/Vendor Open Source Project Can Resolve Tension Between Verticality and Generality

(for both vendors and end users)

- Some people like Vertical Market Platforms
 - ◆ Customers like because
 - Solves unique cross-industry integration problems: industry-specific supply chain issues like trade clearing, bank wire transfers, check clearing, inter-airline reservations, etc.
 - ◆ Small/mid-sized specialist vendors like vertical markets
 - Offer some competitive exclusion against big guys because of their unique knowledge, agility, ability to get into a market first
- Some people don't like Vertical Market Platforms
 - ◆ Startups should like vertical markets but don't
 - VCs often think vertical markets too small to justify investment
 - ◆ Mega-vendors see verticality as unappealing (unless you're talking about padding with services)
 - Addressable market far narrower than core business – even if core business is maturing and growth rate is slowing.
 - Only priced as giveaway to drive for “core” products.
 - Temptation to try to create proprietary “lock in.”

How Eclipse Leverages Convergence of Platform and End Users

- Timing of the opportunity is right
 - ◆ MSFT having trouble driving platform component innovation into customer base
 - Vista
 - Business applications as platform (“Project Green”)
 - ◆ Rise of new programming technologies
 - Stackless Stack a la James Governor of Redmonk (Redmonk quote here)
 - PHP, in particular.
- Shared R&D model makes economics attractive for large vendors to build mix-and-match platforms, and for small vendors to drive verticalization off a common platform base.
- Governance process already in place for managing competing corporate interests
 - ◆ Neutral governance mechanism drives structural agility in the marketplace
 - ◆ Even big vendors can get agility by participating in Eclipse
 - ◆ Enforces loosely-coupled (stackless) architecture, countering tendency to bog down too deep.

Conclusions

- There is a new platform paradigm emerging
 - ◆ To be truly different, it should be designed and built with active customer-vendor participation
- Eclipse is uniquely positioned to lead creation of a new platform
 - ◆ Take advantage of platform fragmentation trend
 - ◆ Governance mechanism in place for broad participation in design and evolution
- Archipelago of platform components is broadest addressable market of all
- Key challenge is to continue to broaden the platform while involving key strategic users on a large scale.

Questions, Praise, etc.

- Sincere Questions & Adoring Praise
 - ◆ In this room, immediately following the presentation
- Brickbats, Complaints, Insults, “Stump the Analyst”
 - ◆ 5:00pm: fast lane, 101 freeway northbound between Great America Parkway and Lawrence Expressway



Eclipse, Open Source, Wall Street and Competition: Big Drama, Big Money

Brent Williams
Managing Director, Equity Research
The Benchmark Co., LLC
bwilliams@benchmarkcap.com
brent_williams@hotmail.com