Open Source Business Models: A Wall Street Look at a Wild 2006 and the Prospects for Even More Fun in 2007

Brent C. Williams
(temporarily) Independent Equity Research Analyst
brent_williams@hotmail.com

More About Me
Than You Really Wanted To Know

- 10 years' experience as Wall Street stock analyst covering software stocks.
- Almost 20 years' experience in the software business before moving to Wall Street
  - Software engineer on multiple platforms, including mainframe, midrange, PC.
  - 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.

What We'll Talk About

- What Recent Events Tell Us About Business Models
- What Investors Like in a Business Model
- Constructing an Open Source Business Model from Scratch
- Technology Trends that Could Affect Business Models
  - Opportunities & Threats
  - Competition from Microsoft and Vista: An Update
- Conclusions

What Recent Events Tell Us About Business Models

Oracle “Clones” Red Hat At Half the Price

- Oracle “clones” Red Hat in October 2006
  - Announces distribution “based on” Red Hat Enterprise Linux
  - Oracle charges half what Red Hat charges for “same” product
  - Enterprise-class support from thousands of trained technicians
- Customer benefits include
  - Full stack from single vendor (just like Microsoft)
  - Savings on high software acquisition costs
  - Simplify life by dealing with fewer vendors

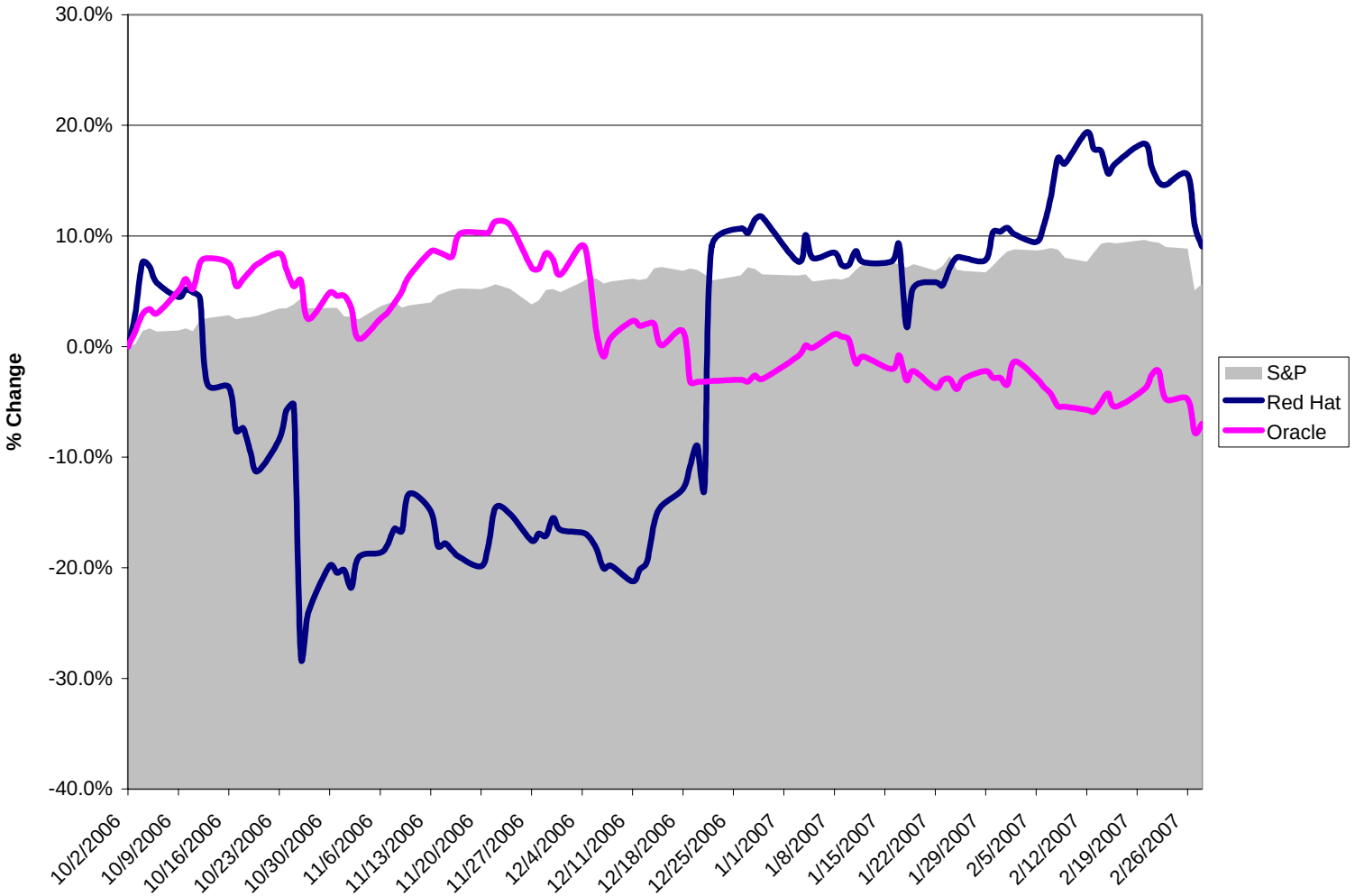
The Success So Far

- First 90 days download statistics
  - 9,000 people download Oracle Linux
  - 1,000,000 people download Fedora Core 6, another “free” Red Hat “clone.”
- A veritable groundswell of support
  - Reference accounts, particularly running non-Oracle software
  - Ecosystem of third-party vendors stampeding to support Oracle Linux

Effect on Red Hat Pricing

- What price changes did Red Hat make immediately in the wake of the Oracle announcement?
  - None. Zero, zip, nada.
  - We're not hearing of any individual deal discounting.
  - Red Hat knows that they have a premium brand, so ignoring people competing on price is the right strategy.
- The role of a premium brand
  - Lamborghini ignores price competition between Hyundai and Kia.
  - Oracle ignores price competition between MySQL and PostgreSQL.
- General Observations on Software Pricing
  - Software is not price competitive at market level, no matter who thinks it is.
  - Software companies *do* discount on initial deals then make it up with higher prices later, but this is hard with subscription model.

Impact on Red Hat Stock Price



What happened here?

- Wrong Economic Model of the Software Business
- Poor product launch -- self-inflicted problems
  - Blogosphere figured out instantly that they are using CentOS
  - Problems installing product; unreliable download site
- No credibility for support offering.
  - No hiring of dedicated support.
  - Sending database support reps to training class won't cut it – didn't work for Novel in 2003 for SuSE launch, either.
- Focus on solving a vendor problem, not a customer problem.
  - Raises indirect costs for switching.
  - Raises "contingent costs" excessively for no benefit
  - Customers fantasize about vendor consolidation but won't pay for it.
- Solving the wrong vendor problem, at that...

Solving the Wrong Problem

- Controlling the entire stack
  - Worked well for Microsoft until recently.
  - Worked for IBM in the 1960s-1980s
    - IBM knows well the limits of this strategy (no applications)
- Getting people off proprietary hardware
  - RAC very successful at this; driving significant portion of growth in database licenses
  - Every dollar not spent on Sun or AIX hardware maintenance can be spent on Oracle maintenance

What's the real problem?

- Mature database business: is there a "Next Big Thing?"
  - For 20 years, growth of database business is due to new features allowing databases to enter bigger addressable markets.
- Application technology is moribund
  - When will double entry bookkeeping be replace by triple-entry?
  - Incremental modules and riding platform shifts are the big opportunities
- Company size makes it hard to sell individual products
  - Makes it hard for a new acquisition to deliver "synergy" (1+1=1.2)
- Integration of acquired products and technologies
  - Risk of Project Fusion is enormous
  - SAP can play conveniently against technology migration risk and expense
- Weak partner channel versus Microsoft and IBM
  - Competing increasingly with partners makes it harder to sign up new ones.
  - "Community" is an important type of partner relationship
- … Cloning Red Hat solves *how* many of these?

(These are problems common to many large software companies, not just Oracle)

Microsoft “Endorses” Novell Linux

- In November, Microsoft and Novell announced a “patent cooperation” agreement that exempted Novell’s Linux customers from patent lawsuits from Microsoft against Linux
  - Deal was carefully worded to avoid collision with GPL2
  - Microsoft is paying Novell more for patent protection than Novell is paying Microsoft
- Microsoft will pay Novell $240 million for discount coupons it can give to its customers to get them to switch to SuSE instead of “other” Linux distributions

Results of the Novell-Microsoft Deal

- Last week, Novell reported $91 million of invoicing for Linux subscriptions, up 650% from prior year
  - $73 million of this was multi-year deals from existing customers still under subscription
  - $18 million in multi-year revenue less than the dollar growth in Red Hat deferred revenue quarter-to-quarter.
  - So are they gaining share?
  - Stock dropped 10% last week on earnings release, despite Linux results.
- Novell management on the defensive
  - Ballmer keeps accusing Linux of IP Infringement, not talking about benefits of Novell deal
  - Novell management has to keep trying to “sell” deal as a good thing

Solving the Wrong Problem, Again

- Novell thinks the problem is catching Red Hat
  - Novell needs to formulate a brand identity for SuSE other than “We’re not Red Hat.”
- Customers aren’t worried about being sued for patent infringement
  - Especially after the SCO lawsuit
- Solving the wrong problem *at the wrong time*
  - Competitive alternatives to Microsoft exist in most of its key markets
    - Sometimes they require a little work, other times, they are “drop in” replacements.
    - Open source is the technology gaining share in all those markets.
  - Open source community can get very good at defending against patent litigation very quickly.
    - Prior art claims, third-party reviews, using Internet to help “patent busters” coordinate efforts.
    - Real possibility that 100% of Microsoft patents will be attacked in initial counterstrike.

Blowback. It's a "beast"

- Reaction from open source community
  - Jeremy Allison, Samba Project lead, quits Novell and goes to Google
  - Novell cancels other open source projects
  - GPL3 strengthened significantly to block similar follies in future
  - Likely increase in shunning of Novell as project contributor because of possible patent "contamination."
- Geopolitical fallout grows
  - Likely to influence course of patent legislation in Europe and other non-US geographies.
  - Likely to result in further tacit or explicit government advocacy of open source.
- Tacitly shows that even Novell believes it can't stop Red Hat itself.
  - Customers recognize the implicit admission of market leadership.

What all the recent fuss tells us about open source business models

- Even people who should know better still don't understand the economics of the software business.
  - Creates opportunities for those who understand the economics.
  - If you solve the wrong problem, you won't get rewarded by the market.
- Trying to "hurt" open source vendors doesn't open up a competitive opportunity.
  - Always stay focused on positive business opportunities.
- Blowback is a "Beast"
  - Blowback will get even more painful for anti-open source misbehavior.

What Investors Like to See In A Business Model

Stock Market 101

- What makes a stock price go up?
  - A growing stream of profits expected in the future.
  - Faster growth in the amount of profits → more value
  - Higher certainty that the company will achieve the expected level of profit → more value

    …ideally, you can do both at the same time.
- What levers can companies pull to drive their stock price higher?
  - More revenue at same operating profit margin percentage
  - More operating margin at same revenue (i.e., lower expenses)
    - Software companies tend to be good at doing both at the same time for structural reasons.
  - ***The big wild card: a higher multiple (i.e., P/E multiple or Enterprise Value/Revenue multiple) that investors will pay for a given level of financial performance (i.e., perceived quality of the future revenue or profit stream).***

Open Source and Profit Margins

Open Source is a great boost to profit margins

- Red Hat estimates that there is $1 billion in R&D powering the Linux ecosystem annually. It gets to take advantage of all of that by investing about $60 million in R&D.
- R&D expense reduction is where most composite vendors see initial benefits of open source.
- A thriving open source community can provide significant benefits in cost to find and fix software defects, especially after production release.

… But cost savings tend to be a one-time boost to stock price; once a new cost savings source is “in the stock,” it loses its ability to move the stock further.
… So revenue generation is the natural place to focus.

How to Generate More Revenue (in General)

- "Economies of scale" type revenue generation
  - Sell product to greater number of customers
    (usually, hiring more sales guys or opening more sales offices)
  - Sell product in larger deals
    (usually, by expanding longtime customer relationships)
- "Agility" type revenue generation
  - Introduce new products faster than competition and win a greater percentage of deals
  - Create and exploit new markets faster than larger competitors, leading to erosion of their value equation

Open Source and Revenue Generation

Open source enables "Revenue by Agility" extremely well:

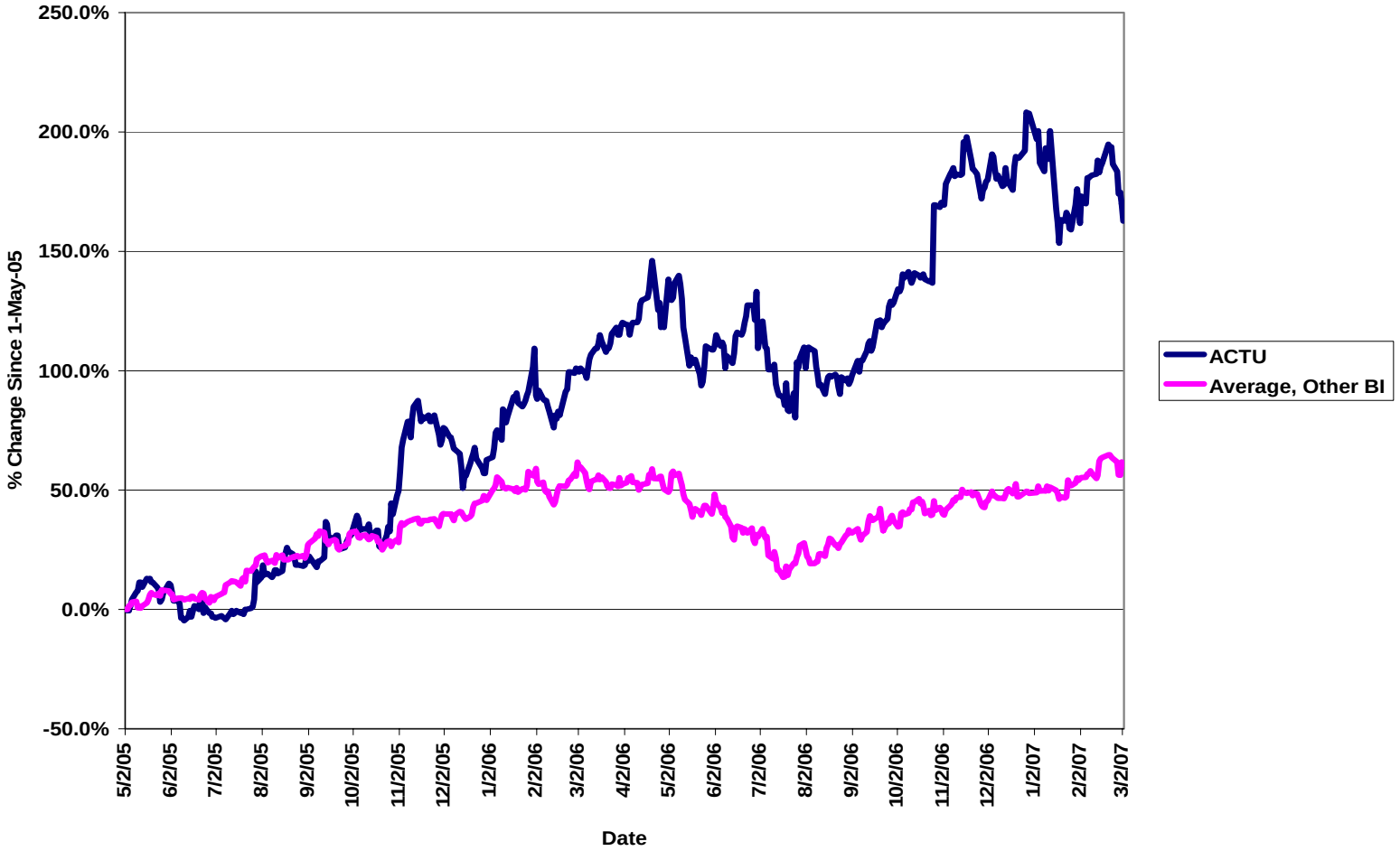
- Open source development model allows companies to create atomic products in existing categories faster
- Open source allows "innovation by integration," fast creation of new products by integration of piece parts from elsewhere, rather than requiring top-to-bottom development project.
- Innovation-by-integration enables new entrants to enter market easily with specialized niche or customized solutions, enhancing value of overall ecosystem and profits for niche vendors.
- New market entrants win, but ecosystem as a whole also benefits because ecosystem can attack new addressable markets (like healthcare), creating incremental opportunities for all ecosystem players.
- Open source process allows companies to commoditize selected bits of functionality that should be commoditized (IDE editors, class libraries) so they can focus on (higher margin) core competencies with less competition

Open Source and multiple expansion

- Show that you are poised to deliver multiple types of revenue upside at lower cost *on multiple product lines* because of your open source involvement.
- Show that open source gives you an architecture for driving more product innovations faster than competition.
- With a broad enough message of the benefits of open source to your product agility, investors will give you credit for products that you haven't even thought of yet (i.e., the Google juggernaut)
- Show that open source enables you to win market share against competitors that you couldn't previously win market share against.
- Success story: Actuate found that BIRT not only generated direct revenue but accelerated the growth of their high-end proprietary product

… In other words, your Eclipse involvement is about a lot more than just the direct revenue you're getting from Eclipse-based products & services.

What Happens When Open Source Business Models Work



Creating An Open Source Business Model

Standard Software Business Model

- Come up with a piece of software
- Figure out what unit of measure to charge for: {seat, server, CPU, year, month}
- Charge as much as you think the traffic will bear
  - Unless this software helps you control another market, then it's free.
- Costs don't occur at same time sale occurs, so there's plenty of time to fix your business model later.
- Lock in (if you can get it) makes repeated customer-unfriendly (i.e., "bad") behavior possible
  - One consequence of aggressively behaving badly is extreme profits: 40%-plus operating margins in mature companies with high customer base lock-in.

Open Source Software Business Models

- Have to make money on every sale (in most cases)
  - Can't discount the license 90% up front, then nail 'em on maintenance next year
- Typically priced by subscription over time since you don't have a proprietary "license asset" to sell up front
- Can still demonstrate enough growth to be attractive to early-stage investors; just have to think that way.

- Beware the Red Hat trap: "We'll charge subscriptions for support, because that's what Red Hat does and it's working for them."
  - Need to be thoughtful about your true core competency
  - Not inevitably going to have to do high volume/low price subscriptions

Intangibles are Part of the Business Model

- Good community behavior is expected
  - Good community behavior is measurable -- bug fixes, etc.
  - Bad behavior becomes widely known and can end up backfiring.
- Customer benefit to everyone of good community behavior is bigger, more viable community
  - That's what drove initial embrace of Windows ecosystem in early 1990s.
  - This drives blowback against "attacks on open source."
- Power of brand can offset perceived commoditization and other effects.

Open Source Gives New Things to Charge For

- Being in the “support-only” business is not a stigma because all open source is in it to some extent.
- Pricing for support can be quite high per unit if you pick the right market and right approach.
- Support looks a lot more value added with a lot more pieces that need integrating.
  - Hits the middle between pre-integrated suite and box ‘o’ parts integrated from scratch by consultants
- Platform and standards create opportunity for add-in providers with deep expertise to gain volume from the platform
  - … without “dining on scraps” problem of Microsoft Visual Studio ecosystem

What All The Recent Fuss Tells Us About Open Source Business Models

- Even people who should know better still don't understand the economics of the software business.
  - Creates opportunities for those who do.
- Trying to "hurt" open source vendors doesn't open up a competitive opportunity.
  - "Protecting'" your business model just calls attention to the problems of that model
  - Always stay focused on positive business opportunities.
  - Customers will increasingly penalize customer-unfriendly behavior.
- Blowback is a "Beast"
  - Blowback will get even more painful for anti-open source misbehavior.
  - Geopolitical consequences: patent revolt, government open source policies, EU fines.
  - Non-US opportunities "more different" and more numerous than before.

Myths of Open Source Business Models

- It's a "weaker" business model than proprietary because there is no lock in.
- Somehow prices must be lower because the customer perceived value of open source equivalent is lower ("they didn't have to build it all themselves, so I'm not going to pay as much").
- Commoditization (Econ 101) will wipe out everyone's profits
- Industry standards will wipe out whatever profits are left after commoditization

What is a Commodity?

- Definition
  - A product where the buyer is unable to distinguish between products from different producers.
  - A black-and-white, universal definition of the product exists.
- Examples
  - Agricultural products: wheat, corn, pork bellies, orange juice, milk
  - Metals: gold, silver, steel
  - Energy: crude oil, electricity, natural gas.
  - Industrial products: nails, plumbing supplies, electrical wire, plywood. Sometimes even DRAM chips.

Structure of a Commodity Marketplace

- Large number of producers and consumers.
- Relative ease for a new producer to enter the market (capital investment, knowledge, licensing)
- Market structure can't be dictated or controlled by any one set of market participants.
- Well-defined product allows risk-free trade with multiple intermediaries between producer and consumer (market structure is flexible and adaptable).
- All acquisition costs are measurable and known: direct, indirect, contingent.

Behavior of a Commodity Marketplace

- No switching costs to buy from different producer
  - Standard product definition,
  - All acquisition costs known
- Market prices are a function of changes in supply and demand.
- Producers can't affect demand, only supply.
  - Farmer Bob can't increase demand for wheat by having a "season end sale."
- Pricing moves quickly to find a point of supply/demand equilibrium.
- "Excess" profits quickly disappear and producer profits revert to the mean of the economy as a whole.
- Lowest-cost producer wins in a commodity marketplace, because he can sustain "excess" profits longer than all other producers.
  - So economies of scale are key to winning.

Software Market vs. Commodity Market

Generic Commodity Marketplace	***Enterprise Software Marketplace***
• No switching costs to buy from different producer	• How many customers alternate database license deals between Microsoft and Oracle, just because?
• Market prices are a function of changes in supply and demand.	• Software supply is always infinite, since it costs zero to print the next copy.
• Producers can't affect demand, only supply.	• Producers in software *create* entire markets (nobody knew they needed a relational database until somebody invented one). • Producers drive their own demand by differentiating their products from competition.
• Pricing moves quickly to find a point of supply/demand equilibrium.	• How often do software companies update their price books?
• "Excess" profits quickly disappear and producer profits revert to the mean of the economy as a whole.	• Operating margins for software companies have always been above the level of the economy as a whole, regardless of industry growth rate. • Software industry operating profits are at highest level ever, relative to the industry's past.
• Lowest-cost producer wins in a commodity marketplace, because he can sustain "excess" profits longer than all other producers.	• Microsoft spends 25% of revenue on R&D. They are *not* the lowest-cost producer. • Everybody is making "excess" profits in the software industry.

Structure of Open Source Software Marketplace -- Is It a Commodity Market?

Generic Commodity Marketplace	*Enterprise Software Marketplace*	*Open Source Software Marketplace*
• No switching costs to buy from different producer	• Customers loyal to incumbent vendors	• Customers loyal to incumbent ***distribution*** vendors
• Market prices are a function of changes in supply and demand.	• Software supply is infinite (zero duplication costs).	• Software supply is always infinite (zero duplication costs)
• Producers can't affect demand, only supply.	• Producers in software *create* entire markets (nobody knew they needed a relational database until somebody invented one). • Producers drive their own demand by differentiating their products from competition.	• Producers create entire markets. • Producers drive their own demand by differentiating their products from competition ***(maybe in slightly different ways)***
• Pricing moves quickly to find a point of supply/demand equilibrium.	• How often do software companies update their price books?	• How often do **open source** software companies update their price books?
• "Excess" profits quickly disappear and producer profits revert to the mean of the economy as a whole.	• Operating margins for software companies have always been above the level of the economy as a whole, regardless of industry growth rate. • Software industry operating profits are at highest level ever, relative to the industry's past.	• Operating margins for **open source** software companies have always been above the level of the economy as a whole, regardless of industry growth rate. • **Open source** software industry profits are at highest level ever, relative to the industry's past.
• Lowest-cost producer wins in a commodity marketplace, because he can sustain "excess" profits longer than all other producers.	• Microsoft spends 25% of revenue on R&D. They are *not* the lowest-cost producer. • Everybody is making "excess" profits in the software industry.	• **Open source vendors may have slightly lower R&D costs than other companies, and may even see marketing benefits as well, on both open source & proprietary offerings.** • **Not sure yet whether** everybody is making "excess" profits in the open source software industry.

So Is It Really a Commodity?

If the software market does not behave like a commodity market, then software cannot be a commodity product.

Method #2 To Tell If Software Is a Commodity

- If software is a commodity, then profits in the software business should fall to the level of the economy as a whole (around 8%)
- The “put your money where your mouth is” test
  - Any vendor that tells you (or your CIO) that the software market is commoditizing should be happy to offer you a discount equal to the number of percentage points that their operating profit margin (i.e., percentage of profit) is above 8%.
  - For example, Oracle’s operating profit margin has lately been between 33% and 38%.

So What Kind of Market is the Software Market?

- High-Risk Industrial Capital Equipment, a variant on standard Capital Equipment
- Standard Industrial Capital Equipment
  - ***High direct purchase cost (big printing presses, locomotives, etc.)***
  - ***High indirect acquisition cost (personnel training, installation time, complex dependencies on other software in environment, extensive hardware purchases)***
  - ***High lifetime operating cost, emphasis on reliability.***
- Added dimension of extreme failure risk (i.e., high "contingent cost")
  - ***Planes crash (can't buy generic jet engine parts on eBay)***
  - ***People die (radiation therapy machines can't run amok)***
  - ***Stocks plummet (last week's software bug in computing the Dow)***
  - ***Companies go out of business (any large Web site failure or airline reservation site crash)***
- Added dimension of not being able to even measure contingent costs accurately
  - ***Fear replaces numbers as decision making criterion – "safe buy" perception***

What Kind of Market is The Open Source Software Market?

- Just as much a high-risk capital good as regular software
- Exhibits aspects of a branded consumer luxury good
  - Strong brand preference – like perfume
  - Bought for abstract reasons
    - I buy cologne to be irresistible to my target audience
    - I don't buy a given scent because it does the best job of making me smell like elderberries (a concrete benefit).
  - "My IDE can beat up your IDE!"
  - "My Linux distro is cooler than your Linux distro!"
- You can have a branded luxury good with commodity ingredients
  - Strong brand preference for mineral water
  - People buying externalities: cute bottle, distribution, image
  - …even if they can't pass a blind taste test half the time.

Do Interface Standards Compress Pricing?

- Hyundai & Lamborghini both support same “interface standard” for layout of driver controls (steering wheel, gas, brake, clutch, gearshift, turn signal, speedometer, tach)
- Hyundai has far greater financial muscle and low cost producer status versus Lamborghini.
- ***“Interface standard” is not “implementation standard”*** (horsepower, body styling, sound system, seat fabric, …)

2006 Lamborghini Murcielago. $279,000

1988 Hyundai Excel, $975

What About Interface Standards *Plus* A Really Nifty Reference Implementation?

- Reference implementation delivers enormous competitive value:
  - R&D cost savings (which should probably be reinvested)
  - Ease of gaining market share against “that other” platform.
  - Eclipse reference platform is in the control of Eclipse members.
- Reference implementation changes rules for success *slightly:*
  - Favors move to more “solution oriented” products (deeper core competency, higher unit prices)
  - Can leverage common infrastructures to create products targeted at clusters of verticals (not horizontal, not vertical, but “diagonal” functionality) giving some combination of volume and specialization.
    - This was the original (good) strategy behind Microsoft’s “Project Green.”
  - Integration assurance becomes an opportunity because platform popularity means wider variety of stuff to integrate; open source enables easy bug fixing, and stable reference port enables focus on integration testing of stuff that matters.
- Echoes the trend in applications – broad “platform” applications now adding deeper vertical functionality by acquiring specialized vendors
- Relatively unlikely to crush pricing unless vendors get complacent outside window of opportunity.

Things That Affect Open Source Opportunities

Competition from Microsoft & Vista

- Redmond is struggling, both technologically and on the business and marketing fronts
  - The “Steve Ballmer Newspaper Reading Quotient”
- Vista underwhelming even our pessimistic assessment of interest and volume
  - Stock took 5% hit on Feb. 16 after downbeat comments from Ballmer
- Suggests that investors are moving back towards growth as an avenue for creating shareholder value.
- Problem of “generic Microsoft port” between XP, Vista, other future versions, as in game world.

Eclipse-as-Platform Opportunity Looking Better Than Ever

- Since last year: OSGI, RCP continuing to evolve; timing continues to look good
- Mobile opportunity also evolving significantly, participation increasing.
  - Smarter and better integrated into corporate environments
  - The post-Blackberry generation”
- File formats becoming a key competitive threat to avoid lock-in
  - ODF is just the beginning
  - Renewed global emphasis on “true” standards because of ODF
- .NET looking harder for vendors and customers to support
  - Complexity without a point

Vertical Market Opportunity is Particularly Interesting

- Suits well the narrower market, higher ticket price business model for Eclipse offerings
- Resonates well with major strategic initiatives from IBM Global Services, other integrators, large application vendors (SAP, Salesforce.com).
  - Customers wary of integrator-led verticalization because of possible “reinventing the wheel” business model.
  - Defined solution sets take away choice to get integration -- not always an acceptable tradeoff.
- Rich opportunities for Eclipse-driven integration solutions, creating significant opportunities for add-in vendors with relevant expertise.
  - This may finally be the catalyst for application frameworks
  - “Project Green” delays, feature cuts create an opportunity for Eclipse
- Opportunity for new subproject creation because of specialized data management requirements.
- Excellent chance for Eclipse to lead this wave because of broad consortium membership base and governance structure to avoid pitfalls of previous standards groups, consortia, etc.

SOA

- Enterprise Level Capabilities
  - Bad news: nobody's going to "go" SOA like they "went" Internet.
  - Good news: eventually, it all goes SOA.
  - Customers starting to discover that SOA isn't easy.
- The Other Half of the SOA Opportunity
  - POX, REST, Ajax, etc. giving simple (mostly non-transactional) services
  - End user developers influence IT strategy and architecture selection
- Can enable secondary class of specialist vendors on top of SOA-enabled transaction suites (like SalesForce.com's AppMarket)

Rich Internet Apps, In General

- AJAX
  - Comes at the perfect time, just as Microsoft says next generation UI requires more complexity (WPF/Avalon)
    - But declarative UI's are nice, and AJAX tools could help
  - Marketing nirvana to draw focus back to client development
  - Great for content-oriented stuff (RSS, structured documents, BI reports, etc.)
  - Not so good (yet?!) for transaction-type documents
  - Need to avoid temptation to get too complex
- Need to provide layer to integrate between "applications," like RCP applications (i.e., cut and paste)

Open Source Business Models: A Wall Street Look at a Wild 2006 and the Prospects for Even More Fun in 2007

Brent C. Williams
(temporarily) Independent Equity Research Analyst
brent_williams@hotmail.com