Models of Software Acceptance

How Winners Win

Richard P. Gabriel
Stanford University

How Is Technology Accepted?

Building business success on technology is not easy—myths abound based on common sense, tales told by those who have won, analogies to things like evolution, and appeals to inventiveness and innovation. When we look closely at how technology is accepted and how success is built on it, the picture is quite different, and the process of acceptance is both lengthy and unpredictable. In this talk we'll look at the myths and the realities, we'll look at many specific examples, and we'll conjecture a set of principles that might work.

Common Business Misconceptions:

- First-to-market wins
- Inventiveness is good—because evolution tells us so
- Best technology wins
- Best marketing wins

First-to-Market Wins?

- Does the first company to market always win?
  - most economists and business theoreticians seem to believe so:

    Over the past decade, a lot of research has reinforced the commonsense view that "first movers"—firms that are first into a market—will end up dominating it. One study suggested (with curious precision) that, across a wide range of industries, market pioneers end up holding a 30% share of their markets, compared with 13% for latecomers. Another found that about 70% of today's market leaders pioneered the markets they dominate.

    Two of today's most influential management theorists, C.K Prahalad and Gary Hamel, have lent their authority to this position. First movers, they argue, can reap benefits in lots of ways: by establishing a monopoly, however fleeting (as Sony did with the Walkman); or by setting standards (as Microsoft did with DOS); or by dictating the rules of the game (Wal-Mart and hypermarkets). The losers, on the other hand, are consigned to a treadmill of catch-up and cost-cutting. [Economist, March 16–22, 1996]

The First May Not Last

- The first company to market almost always succeeds, unless that company goes out of business or drops out of that market . . .

> *[P]revious studies are based on surveys of surviving companies and brands, thereby excluding all the pioneers that failed. This helps some companies to look like pioneers even when they were not. Procter & Gamble (P&G) has boasted that it created America's disposable-nappy (diaper) business. In fact a brand called Chux had been launched a quarter of a century before P&G entered the market in 1961.*
>
> *Rather than using existing databases, Mr Tellis and Mr Golder pored over reports written at the time that each individual market was evolving. This meant studying 50 different markets, each defined to mirror the way consumers think of product categories (eg, "microwave ovens" or "light beers"). Great care was taken to avoid hindsight when defining firms as pioneers. The goal was to answer two simple questions: how did a company appear at the time it entered a market? And how did it then fare?*

The First May Not Last

- . . . and that happens about 90% of the time:

  > *On this measure, the failure rate for pioneers turned out to be high: almost half did not survive. In only 11% of the markets surveyed were today's leaders true pioneers—and the average market share of the survivors is now a mere 10%. The authors argue that the real success goes to "early leaders"—firms that entered the market an average of 13 years after the pioneers and now have a market share three times the size of the pioneers.*

The First May Not Last

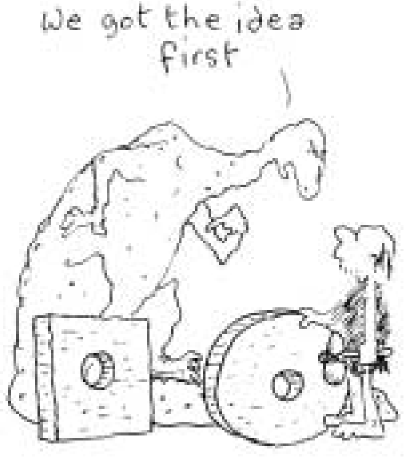
One reason for the success of early leaders is that pioneers often fail to conjure up a mass market. When a product is first launched, its quality is often low, its price high and its applications limited. The personal computer fell into this category, until software firms developed spreadsheets and word processors.

Much the same happened in the case of the video recorder (VCR). America's Ampex, which pioneered the technology in 1956, charged about $50,000 for its early models and sold only a few. It made little effort to cut costs and expand its market. Japan's Sony, JVC and Matsushita, by contrast, saw the potential for mass-market sales and set out to make a video recorder that would cost $500—a goal that took them 20 years to achieve.

In many cases, including Ampex's, the first mover was content to have pioneered the technology, believing its breakthrough was enough to bring market leadership. Micro Instrumentation and Telemetry Systems invented the PC in the mid-1970s, but ceded market leadership to latecomers (such as Apple Computer and IBM) that invested heavily to turn the PC into a mass-market product.

The First May Not Last

The two academics note that ***success in a business related to a new market—and the financial strength that comes with this—is often the reason that early leaders beat the pioneers****. In the early 1960s America's Royal Crown was a pioneer in the consumer market for diet colas, a product that had previously been sold only to diabetics. But PepsiCo and Coca-Cola were able to use their vast financial muscle in other parts of the cola market to crush Royal Crown, despite their late arrival. Indeed, it took Diet Coke only a year to establish market leadership after Coca-Cola launched it in 1983.* [Economist, March 16–22, 1996]



Inventiveness is Good?

- Ampex (first VCR) [*invention*]
- Multics, VMS, Apollo Domain lost [*invention*]
- DOS, Windows 3.1 won [*innovation*]
- Frame, Tex lost [*invention*]
- Word, HTML won [*innovation*]

Best Technology Wins?

- Apollo lost to Sun
- Symbolics lost to C/C++ machines
- Macintosh second to WIntel and losing fast
- Emacs lost to VI
- Lisp/CLOS, ML, Scheme, Haskell lost or are losing to C/C++/Java
- Beta lost to VHS
- NeXT, Taligent, Kaleida, HAL, Thinking Machines all lost

Best Marketing Wins?

- Infinity—expensive, ubiquitous, mysterious advertisements prior to launch
- NeXT—Steve Jobs is the ultimate marketing machine, but NeXT is gone
- Go—continued hype followed by collapse
- Apple (compared to Microsoft)

But, Evolution Says It Ain't So, Right?

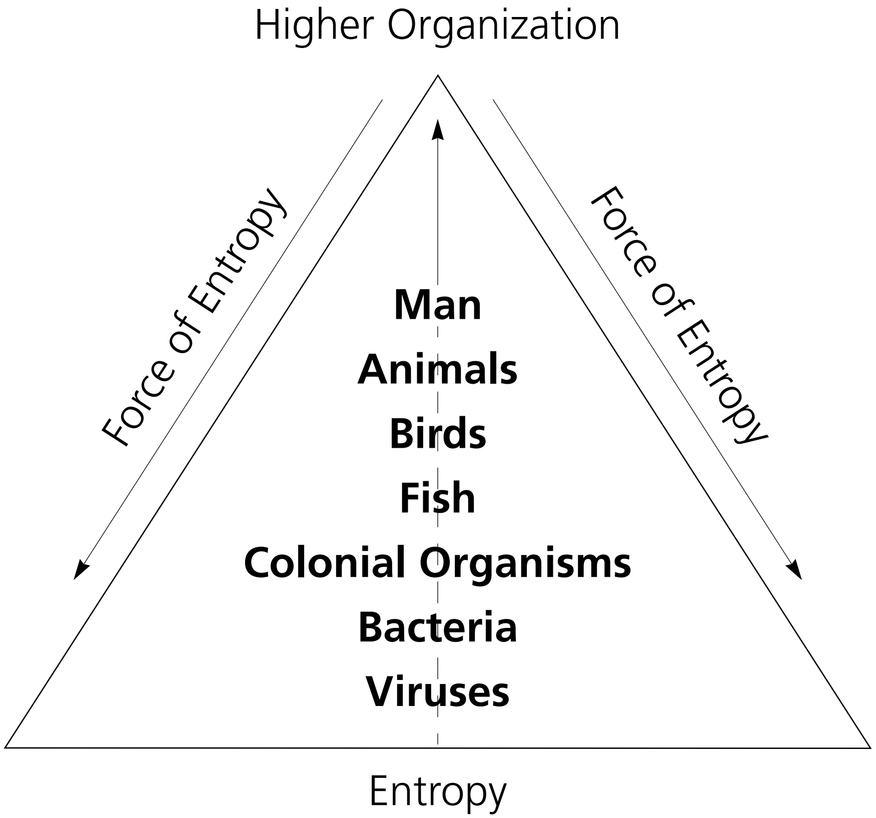
[T]he forces of capitalism are much like those of evolution: the fittest survive. And when the fittest leap forward in their abilities, many weak or sick competitors are driven off or eaten. The benefits are clear: Consumers such as you and me have better tools. [Andrew Binstock, UNIX Review, 1994]

Wrong!

Evolution is the Proper Model for Business Success, Right?

- Evolution inevitably creates progress and increased complexity in life, right? ***Wrong!***
- Natural selection causes improvements, right? ***Wrong!***

Evolution=Progress?



Higher Organization

Force of Entropy

Force of Entropy

Man

Animals

Birds

Fish

Colonial Organisms

Bacteria

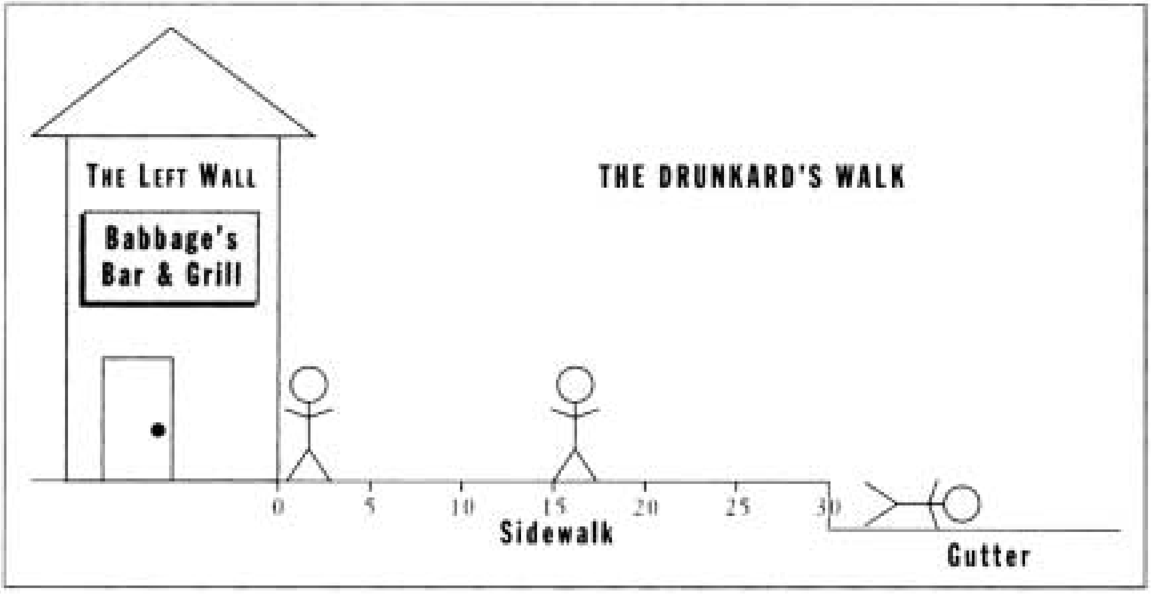
Viruses

Entropy

Evolution=Progress?

Many reversals have occurred along the way, but the overall average across the history of life has moved from the simple and few to the more complex and numerous. During the past billion years, animals as a whole evolved upward in body size, feeding and defensive techniques, brain and behavioral complexity, social organization, and precision of environmental control. . . . ***Progress, then, is a property of the evolution of life as a whole by almost any conceivable intuitive standard****, including the acquisition of goals and intentions in the behavior of animals. It makes little sense to judge it irrelevant. Attentive to the adjuration of C. S. Peirce, let us not pretend to deny in our philosophy what we know in our hearts to be true.* [E. O. Wilson, 1992]

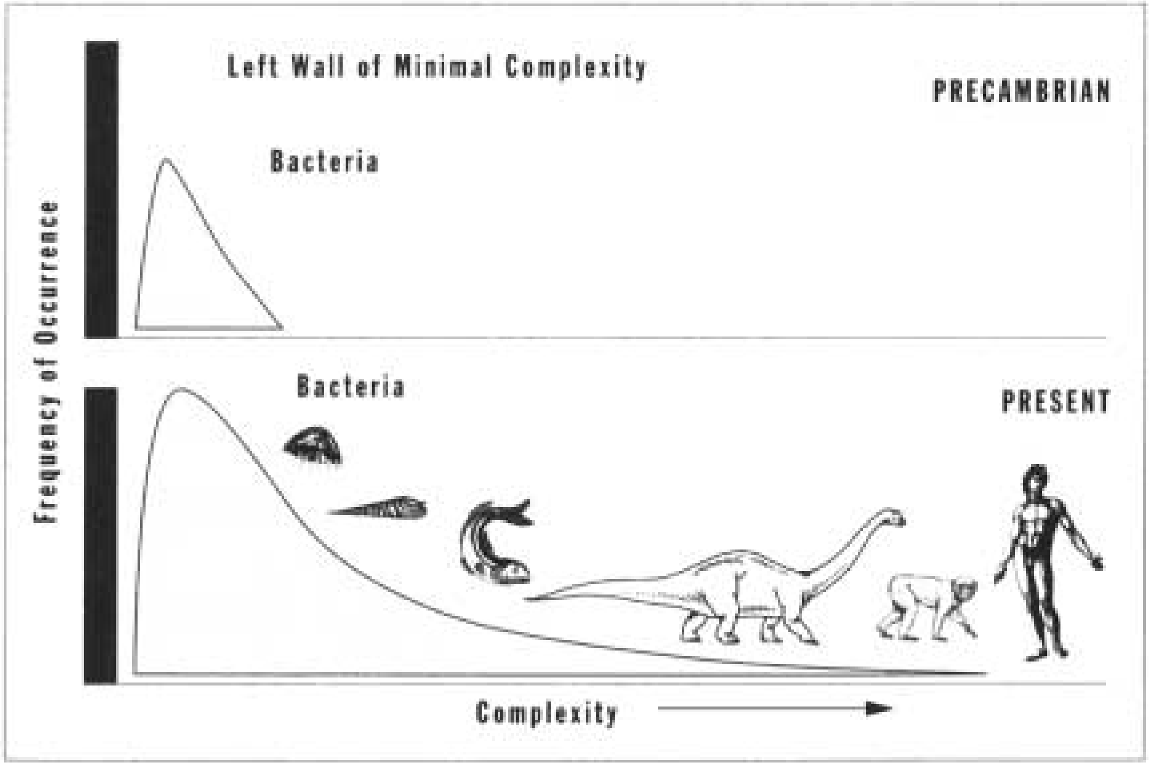
Evolution=Progress?



Evolution=Progress?

- Life begins at the left wall of minimum complexity
- Stability throughout time of the bacterial mode
- Life's successful expansion must form an increasingly right-skewed distribution
- The myopia of characterizing a full distribution by an extreme item at one tail
- Causality resides at the wall and in the spread of variation; the right tail is a consequence, not a cause
- The only way to smuggle progress back into such a system is logically possible, but empirically false at high probability
  - There is no evidence that a lineage that starts away from the left wall demonstrates any directional bias (neither in size nor in complexity)

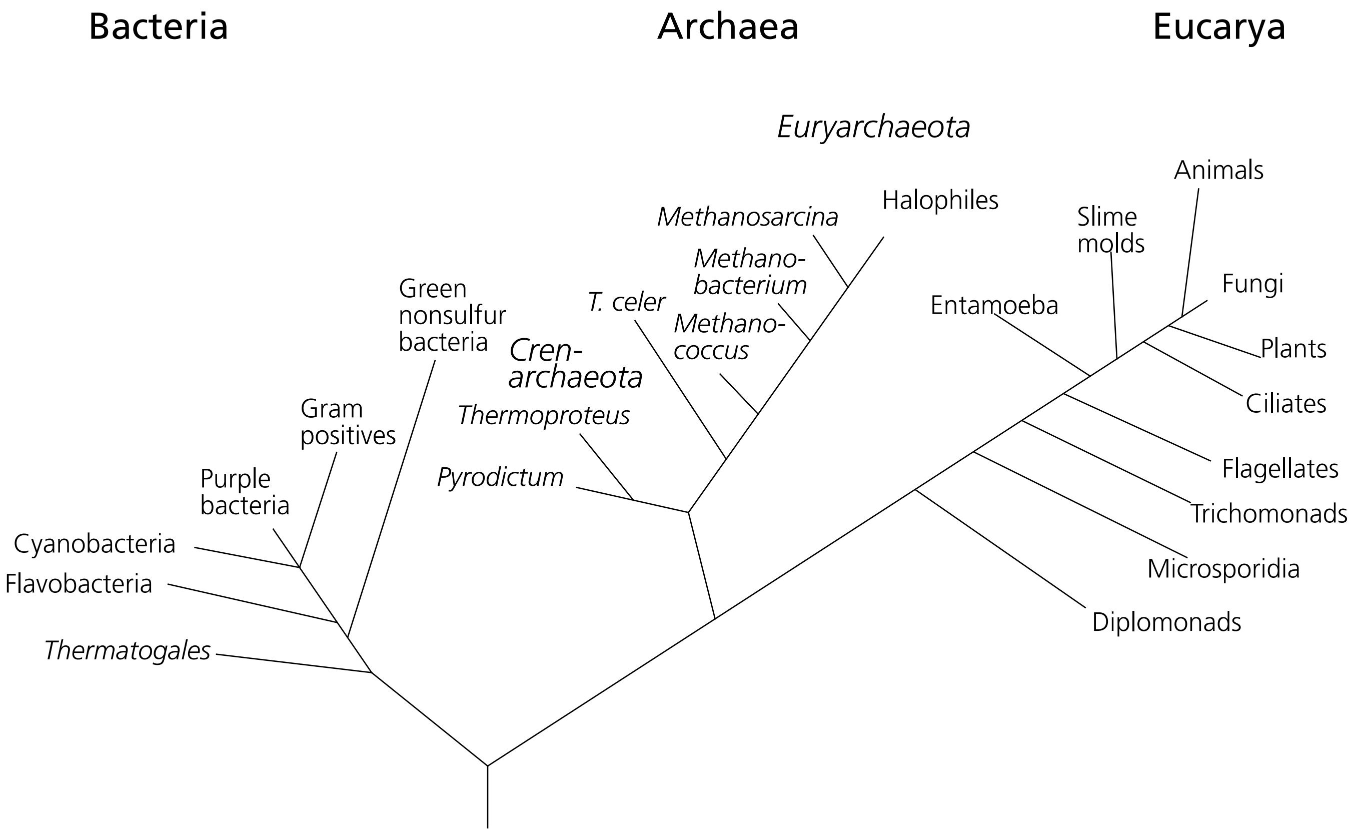
Evolution=Progress?



Evolution=Progress?

- Bacteria appeared 3.5–3.6 billion years ago
- Most of the species today are bacteria
- Bacteria are adapted to every environment (subzero temperatures to 480°F, several miles underground, pH of 1 or 2)
- Possibly, most biomass is bacteria

Evolution=Progress?



At the Molecular Level, Natural Selection Prevents Change

- At the molecular level, changes happen at the maximal (random) rate among genes when those changes are neutral to the organism as a whole. [Motoo Kimura]
- The more irrelevant the organ or structure, the faster its genetic makeup changes:

  Only things that are relatively worthless change rapidly and dramatically

  - biologists use mutation rate to determine utility

Natural Selection Prevents Change

- Brains and wings, for example, cannot be used for their current purposes until they have certain characteristics: e.g. minimum size and complexity. Therefore natural selection did not prevent them from changing with respect to those characteristics. Natural selection preserved the basic structure for their original (found) purposes—cooling and memory, attractiveness to mates (signs of health).
- **Then a change in the environment happened: A capability became important—perhaps by the coincidence that the structure happened to serve an critical function in the new environment—and natural selection:**
    - **selected the changes by preventing their alteration**

Natural Selection Prevents Change

- Some business theorists (or pundits) point to Darwinian evolution as a reason why the free-market system works, but Darwin was under pressure to show the industrial revolution was beneficial by showing it mirrored the natural world:

  > *To respond to the pressure he was facing, Darwin came up with his* ***wedge theory****, which stated that the panorama of life was chock full of species occupying every ecological niche. Each species was like a wedge, and the only way that a new species could find a place in the world was to wedge out another one, and the way that happened was by being better somehow.* [Gould]

- That is, evolution looks like the free-market because the free-market pressured Darwin to make it look that way.

Natural Selection Prevents Change

*Dinosaurs and mammals overlapped by 100 million years, plenty of time for any wedge-related competition to have increased the mammals' niche or even forced out the dinosaurs if the mammals were so much better than dinosaurs. However, it took a disaster—currently believed to be a comet or asteroid hitting the earth—to dislodge the dinosaurs. The small mammals were able to survive the cloud-induced winter or the flash-induced heat that resulted but the dinosaurs were not. The mammals had something that enabled them to survive, and almost certainly it was something marginal or irrelevant until the catastrophic event—**otherwise you have to believe that evolution operates by the future influencing the past**.*

Natural Selection Prevents Change

- Natural selection prevents change by choosing what survives, which is almost always what has survived before because environmental change is slow
- What is free to change is not crucial to survival

> *The most important and useful of these predictions involves a paradox under older Darwinian views. If selection controls evolutionary rate, one might think that the fastest tempos of alteration would be associated with the strongest selective pressures for change. Speed of change should vary directly with intensity of selection. Neutral theory predicts precisely the opposite—for an obvious reason once you start thinking about it. The most rapid change should be associated with unconstrained randomness—following the old thermodynamic imperative that things will invariably go to hell unless you struggle actively to maintain them as they are. After all, stability is far more common than change at any moment in the history of life. In its ordinary everyday mode, natural selection must struggle to preserve working combinations against a constant input of deleterious mutations.* ***In other words, natural selection, in our technical parlance, must usually be "purifying" or "stabilizing."*** *Positive selection for change must be a much rarer event than watchdog selection for tossing out harmful variants and preserving what works.* [Gould]

The Free Market Follows the Revised Evolutionary Model

- Everything is stable until the environment changes
- Already-existing technology is quickly adapted
- After the change, companies improve/innovate slowly to maximize ROI

  The free market means improvement for the consumers, but ***at the slowest possible rate****, and companies that try to go faster than that rate are almost always hammered down or killed* [Gabriel]

  - they spend too much compared to the competition and cannot charge for the differential development costs
  - higher cost and greater risk (too far from the left wall) cause low demand

Technology Lifecycle

Technology is accepted slowly by large mainstream customer segments:

- Technology in the lab (*t=0*)
- Technology in the first company (*t=2–10 years post-lab*)
- Technology in the first successful company (*t=5–20 years post-lab*)
- Technology acceptance (*t=10–25 years post-lab*)

Example: Window Systems

- First window system: Stanford/SRI/Xerox PARC ~1975, MIT ~1976
- First commercial use: Symbolics (1979), LMI (1979), Xerox Star (~1980), Apple Lisa (~1982), Apple Macintosh (1984)
- First use by a successful company: Microsoft (~1989)
- Technology acceptance: Microsoft (~1995)

Example: Workstations

- First workstations: Xerox ~1973, MIT ~1974
- First commercial use: Apollo ~1980
- First use by a successful company: Sun 1984
- Commercial acceptance: Sun, HP, IBM 1990–1992 (use in corporate client-server applications)

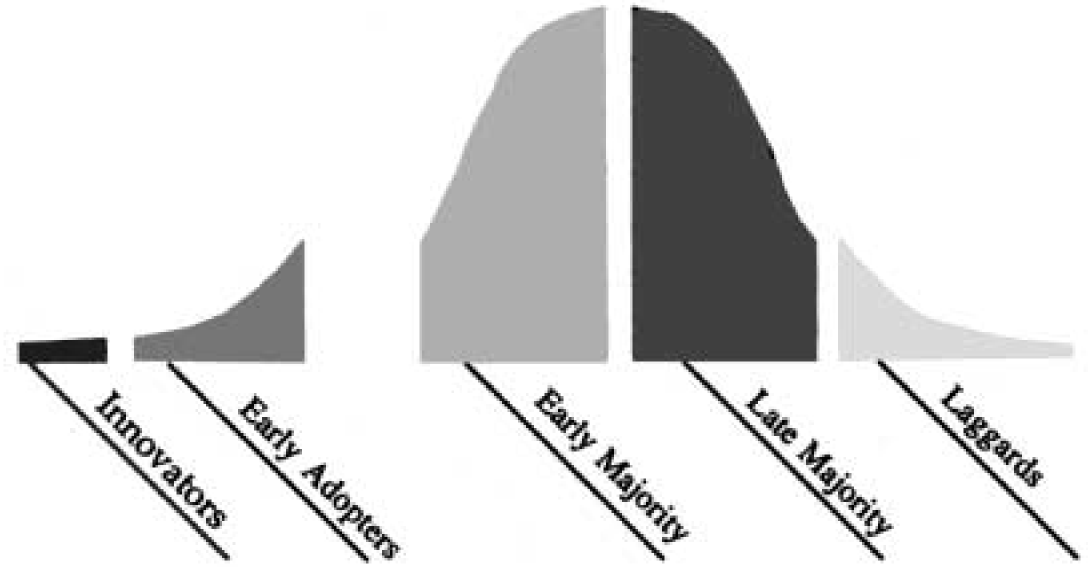
Example: VCRs

- First VCR: ~1954
- First commercial use: Ampex 1956
- First use by a successful company: JVC, SONY ~1975
- Commercial acceptance: Sony, JVC, etc. ~1985

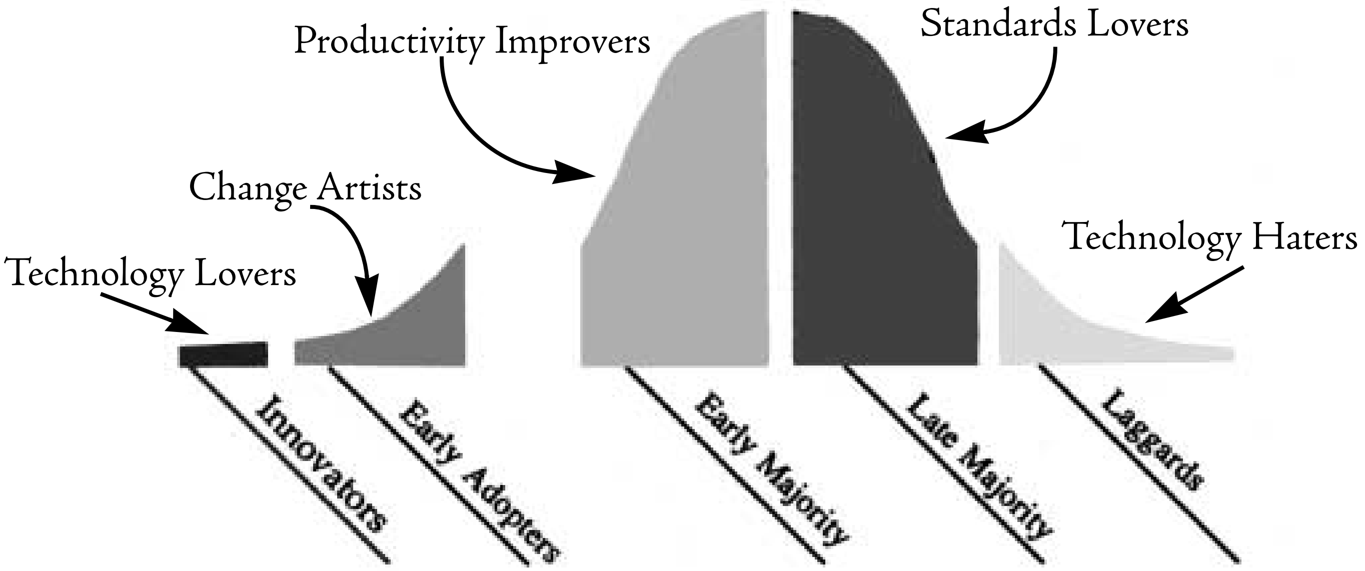
Example: Spreadsheets

- First spreadsheet concept: IBM ~1965 (Sketchpad)
- First spreadsheet in spreadsheet format: Bricklin/Frankston ~1977
- First commercial use: Visicalc ~1978
- First use by a successful company: Lotus ~1983
- Commercial acceptance: Lotus, Microsoft, Borland ~1986

The Elusive Chasm Explained



The Elusive Chasm Explained



Productivity Improvers

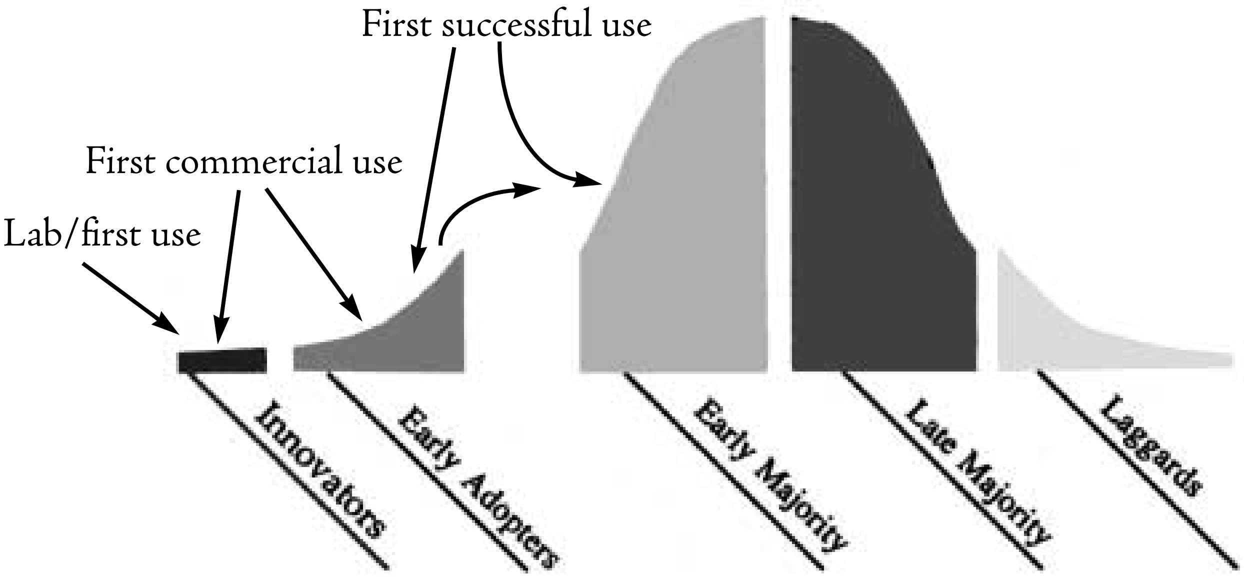
Standards Lovers

Change Artists

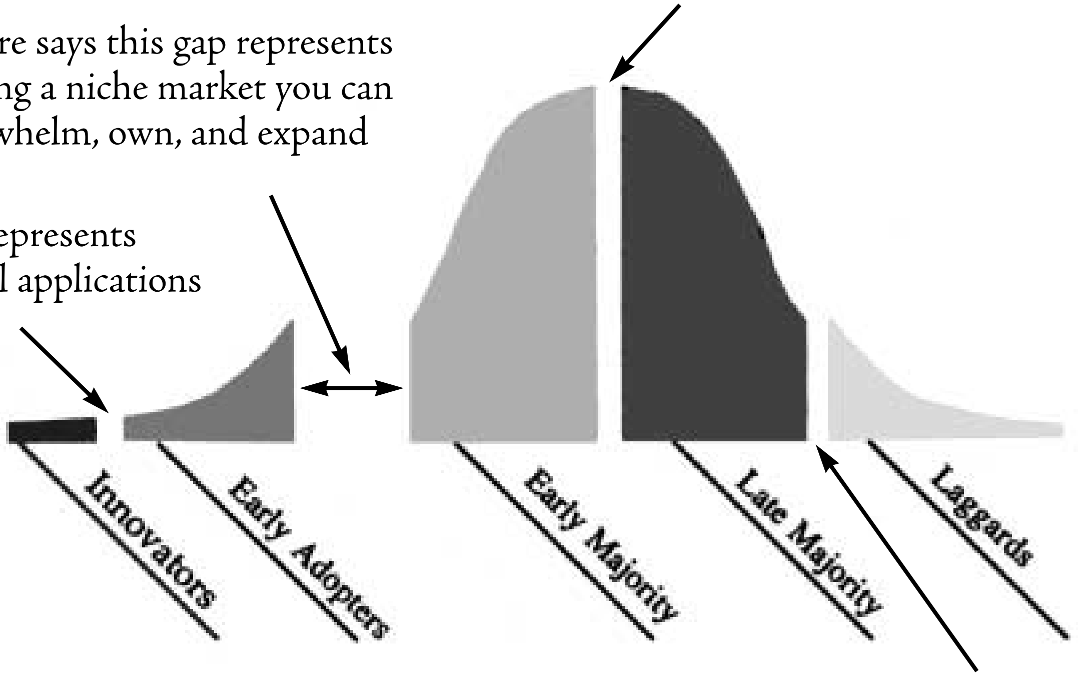
Technology Lovers

Technology Haters

The Elusive Chasm Explained



The Elusive Chasm Explained



This gap represents an established standard practice in which technology is hidden

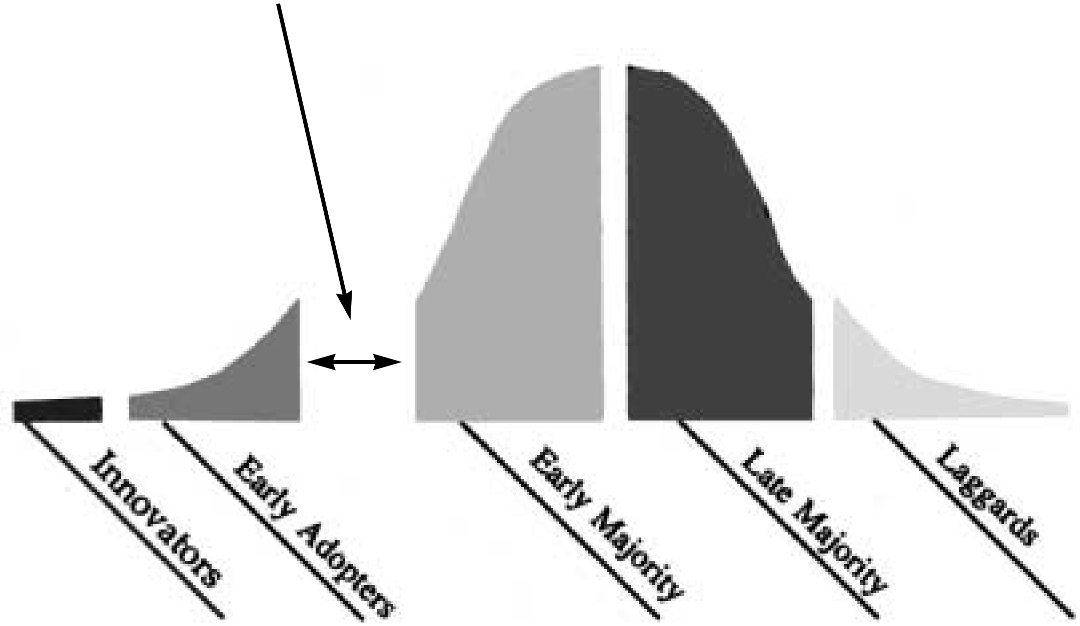
Moore says this gap represents finding a niche market you can overwhelm, own, and expand

This gap represents finding real applications

This gap represents the consumer having no choice whether or not to use the technology

The Elusive Chasm Explained

This gap really represents a change in the environment that renders the technology useful/desirable/hot/necessary. It turns a vitamin pill into a pain killer, a nice-to-have into a must have.



Often the seeds for the environmental change are planted by the pioneer who plants the idea that a technological solution is possible: **How necessary to the process is the failed pioneer?**

The Elusive Chasm Explained

VCRs:

- Only broadcasters could use VCRs in the '50s and '60s
- The technology becomes inexpensive enough for the mass market ('70s)
- The culture changed (mid '80s)
  - TV watching is prevalent
  - time-shifting needed because people are too busy
  - full-length movies are too expensive

This is the environmental change that made VCRs useful rather than irrelevant

The Elusive Chasm Explained

Spreadsheets:

- Only IT/MIS departments could use spreadsheets in the '60s
- The technology (hardware and software combined) becomes inexpensive enough for the mass market—Apple II (late '70s)
- The environment changed (mid '80s)
  - a respected computer maker with a fast-enough, large-enough, affordable-enough personal computer—IBM
  - small companies and startups sprouting to supply an eager demand—a strong entrepreneurial climate
  - people in departments accustomed to using computers—home computers, experience at school, and word processors

Toward a Theory of Technology Acceptance

- Choosing technology
- Characteristics of product design
- Characteristics of product implementation
- Characteristics of the surrounding environment

Acceptance Models

Explosive Growth

- **Virus/Symbiont Model:** Attach to something growing [*opportunistic*]
  - Unix (to PDP-11's, Vaxes), C (to Unix), DOS (to IBM PC)
- **Game Model:** Be way fun ("my manager isn't looking, maybe I'll play!"), then assert serious use (information dissemination/Intranets) [*aggressive*]
  - Netscape Navigator, spreadsheets
- **Evolutionary Model:** Become useful because the environment has changed [*opportunistic*]
  - Java, VCR's
- **Inexorable Growth Model:** Gradual acceptance through features [*passive*]
  - Illustrator, Photoshop

Inexorable growth followed a small environmental change and a small game-like puff of smoke

The Game Model and Netscape

- The Game Model, when it works, creates the environmental change needed for explosive growth
- 38 million copies of Navigator
  - 30 million copies of Excel
  - 21 million copies of Word
- Navigator
  - displays text crudely compared to any other "publishing engine"
  - displays graphics crudely compared to any other "graphics engine"
  - does not help find anything on the Net (you need an URL or a search engine)
  - uses one of the slowest transfer media (the Net)
  - frequent failures (the Net and servers)
  - originally didn't help with e-mail, didn't edit text, did no computations
  - uses one of the worst markup languages (HTML)
  - provides a crude user interface

In a world containing Windows-95, Macintosh, Frame, Word, Tex, X, how could such a thing possibly have succeeded?

Have You Ever Played Myst?

- Places to go
- Things to see
- Exploration—curiosity
- Diversion—stress release
- Killing time—boredom

By the time Netscape shipped, there were enough home pages around the world created by those with not enough to do that **surfing the web became a game**.

What Are the Supposed Benefits of the Web?

- Everyone knows it (because 38 million people have played the game)
- Universal user interface (it's a popular game)
- Simple to implement (trivial protocols)
- Platform independent platform (simple server implementations put it everywhere)
- Revolutionizes client/server (fat client thins, client is already everywhere)
- Easy to use (it's a popular game)
- Intranet computing
  - internal information sharing
  - interdepartmental applications
- Internet computing
  - FedEx
  - customer care applications
- The "benefits"—information dissemination and simple client/server applications—were demonstrated in the Web Game by people with hobbies (the Mastiff homepage, for example)

In Case You Wondered . . .

- Hypertext invented in the late '60's by Ted Nelson
- Hypertext systems in modest use in the late-'70's
- Hypertext systems in common use in the '80's
- Markup languages invented in the '60's
- Markup languages in common use in the '70's
- Wide-area network invented in the late '60's
- National network in use in the early '70's (ARPANet)
- International network in common use in the late '70's
- Web first developed in the mid-'80's for "publishing" scientific results (in competition with Gopher)

Whither Netscape

- Netscape created an environmental change and was therefore a successful pioneer
- But recall Tellis and Golder:

> *One reason for the success of early leaders is that pioneers often fail to conjure up a mass market. When a product is first launched, its quality is often low, its price high and its applications limited.*

- Netscape failed to conjure a mass market, just a mass of users, and it now lacks the capital to withstand a run at it by Microsoft, who has the wherewithal and a monopoly position to exploit

Toward a Theory of Technology Acceptance: Technology

- Choose technology that is at least 10 years out of the lab, preferably 20 years
- Choose a technology that has already had an unsuccessful or minimally successful commercial application, perhaps in a high-end narrow niche—this stimulates later acceptance, the *I-heard-of-that-before* phenomenon
- Choose a technology that appeals to a market need, though not a market requirement
- Make the smallest investment in the technology to address adequately market needs, then plan on slow, slow, inexorable progress
- Choose a technology to be at the core of your business, but fill out the complete product with the work of others: Use **Wade's Maxim**:

No one ever made money by typing

Toward a Theory of Technology Acceptance: Design

- ***Simplicity:*** The design should be simple in implementation. The interface should be simple, but anything adequate will do. Implementation simplicity is the most important consideration in a design.
- ***Minimal Completeness:*** The design should cover only necessary situations. Completeness can be sacrificed in favor of any other quality. In fact, completeness should be sacrificed whenever implementation simplicity is jeopardized.
- ***Minimal Correctness:*** The design should be correct in all observable aspects.
- ***Minimal Consistency:*** The design should be consistent as far as it goes. Consistency is less of a problem because you should always choose the smallest scope for the first implementation.
- ***Minimal Abstraction:*** The design should require limited abstraction competence. Further, abstraction gets in the way when it makes implementation too hard, too slow, or hides information that shouldn't be hidden. (And . . . **Abstractions = page-faults**)

Toward a Theory of Technology Acceptance: Implementation

- The implementation should be fast
- It should be small
- It should interoperate with the programs and tools that the expected users are already using
- It should be bug-free, and if that requires implementing fewer features, do it

Toward a Theory of Technology Acceptance: Environment

Do one of these:

- Choose the inexorable growth model and hope for the best
- Look for some other product that is growing and become part of it
- Look for an application of the technology or product which is like a game
- Look for an environmental change and take advantage of it:
  - culture
  - computing practices
  - business focus
  - business pressure

If your **existing** technology cannot be adapted to meet the challenge of the environmental change, it cannot have much effect, so don't waste resources.

Look for a product or niche where a larger competitor (in a related market) cannot beat you using Gambler's Ruin

Additional Requirements for Language Acceptance

Languages are accepted and evolve by a social process, not a technical or technological one.

- A successful language must have modest or minimal computer resource requirements
- . . . must have a simple performance model
- . . . must not require users to have "mathematical sophistication"
- . . . must be a minimally acceptable language
- . . . must be similar to existing popular languages
- . . . must be available on a wide variety of hardware
  - the most prevalent hardware is 2–3 years behind the current state-of-the-market
  - work toward the "bacterial mode"—don't be fooled by means or medians
- It helps to have local wizards or gurus for the language

How the Theory Works

- You have related, undervalued technology that is:
  - untested in the market
  - irrelevant or harmful in your market
  - engineering-mature but relatively unknown in the market
- The environment has changed so there is high demand for products easily developed from your technology base
- You quickly create a minimal product using *worse-is-better* design and implementation criteria, perhaps in a vertical niche market
- The product takes little development time, so it is out early and can be adopted as the de facto standard in a new market area
- It is implementationally and functionally simple, so it can run on the smallest computers. Maybe it can be easily ported as well—if it uses a simple portability model. At any given time the mainstream computer users—whether individuals or corporations—are running hardware at least two generations old.
- If it has some value, it will be spread like a virus or a game
- If it has value and becomes popular, there will be pressure to improve it, and over time it will acquire the quality and feature-richness of systems designed another way, but with the added advantage that the incremental features will be those the customers or users want, not those that the developers think customers should want.

The Theory Applied to Dynamic Languages

Lisp (circa 1989):

- Technological model:
  ✓ Old technology: in the lab from 1959 through 1980
  ✗ Tried before: no
  ✓ Appeals to market need: yes, AI fury
  ✗ Smallest investment: no
  ✗ Minimize typing: no
- Design model:
  ✗ Simplicity: no
  ✗ Minimal completeness: no
  ✗ Minimal correctness: no
  ✗ Minimal consistency: no
  ✗ Minimal abstraction: no
- Implementation:
  ✗ Fast: no
  ✗ Small: not yet
  ✓ Interoperate: sort of
- Environment:
  ✓ Cultural change: yes (AI)
  ✓ Quickly adapted: yes
  ✓ Gambler's Ruin-safe: yes

The Theory Applied to Dynamic Languages

Lisp:

- Language requirements:
  ✓ Runs everywhere: yes
  ✗ Minimal computer requirements: no
  ✗ Simple performance model: no
  ✗ Minimal mathematical sophistication: no
  ✓ Minimally acceptable for purpose: sort of
  ✗ Similar to popular language: no
  ✓ Gurus: yes

Verdict:

Lisp helped AI fail and so it died out for a while. Lisp never got beyond the pioneer stage. The environment was right to leap the chasm, but failure to execute overwhelmed the pioneers.

Guilty of blowing a great chance.

The Theory Applied to Dynamic Languages

Smalltalk:

- Technological model:
  - ✓ Old technology: in the lab from 1967 through 1980
  - ✗ Tried before: no
  - ✓ Appeals to market need: yes, business reengineering and client/server
  - ✗ Smallest investment: no
  - ✗ Minimize typing: no
- Design model:
  - ✓ Simplicity: yes
  - ✗ Minimal completeness: no
  - ✗ Minimal correctness: no
  - ✗ Minimal consistency: no
  - ✓ Minimal abstraction: sort of
- Implementation:
  - ✗ Fast: no
  - ✗ Small: no
  - ✓ Interoperate: sort of
- Environment:
  - ✗ Cultural change: yes (business modeling, OOP), but this cultural change is fading away fast
  - ✓ Quickly adapted: yes
  - ✗ Gambler's Ruin-safe: no

The Theory Applied to Dynamic Languages

Smalltalk:

- Language requirements:
  ✓ Runs everywhere: yes
  ✗ Minimal computer requirements: no
  ✓ Simple performance model: sort of
  ✗ Minimal mathematical sophistication: sort of, but the syntax is too weird
  ✓ Minimally acceptable for purpose: yes
  ✗ Similar to popular language: no
  ✓ Gurus: yes

Verdict:

Pioneers still lead whatever charge is left, but a late arrival became the early leader (IBM)—Gambler's Ruin says this competitor should win. Smalltalk made better technological choices than Lisp.

The jury was out a long time, but Smalltalk followed the lead of the Lisp companies too much (by accident) and this language is now marginal. IBM has moved on to Java.

The Theory Applied to Dynamic Languages

Java:

- Technological model:
  - ✓ Old technology: Sort of, modules and interface model in Lab only 8 years ago
  - ✓ Tried before: yes, C++
  - ✓ Appeals to market need: yes, Web mania
  - ✗ Smallest investment: originally yes but changing rapidly
  - ✓ Minimize typing: yes, relying on others for JIC, OE, windowing
- Design model:
  - ✓ Simplicity: yes, except for C++-like syntax; but rapidly becoming complex
  - ✓ Minimal completeness: yes
  - ✗ Minimal correctness: no, too elegant
  - ✗ Minimal consistency: no, too elegant
  - ✗ Minimal abstraction: maybe not
- Implementation:
  - ✗ Fast: no
  - ✗ Small: no, size is sometimes hidden by browser, but not in applications
  - ✓ Interoperate: yes, so far
- Environment:
  - ✓ Cultural change: yes (Web game)
  - ✓ Quickly adapted: yes, from earlier attempts to apply Oak to application areas
  - ✗ Gambler's Ruin-safe: not really

The Theory Applied to Dynamic Languages

Java:

- Language requirements:
  ✓ Runs everywhere: in theory, yes, but actually having problems
  ✗ Minimal computer requirements: originally yes, but rapidly losing this advantage
  ✓ Simple performance model: sort of
  ✗ Minimal mathematical sophistication: maybe not
  ✓ Minimally acceptable for purpose: yes
  ✓ Similar to popular language: yes, but also to unpopular ones
  ✓ Gurus: yes

Verdict:

A large second generation player (Sun) leads the charge, but newcomers abound, especially a large one (Microsoft)—Gambler's Ruin says Microsoft wins. This language made much better technological choices than either Lisp or Smalltalk. The environmental change (Web, Web, Web) is unprecedented in magnitude. Java stole Smalltalk's environmental advantage (C/S).

Lisp, Smalltalk, and C++ were the pioneers for the OO and dynamic nature of Java.

Netscape-style plug-ins could provide a means for non-Java solutions to win, such as Lisp or Smalltalk. Except for modules and interface types (a dangerous bet), Java is old technology—both as a language and in its application to the Web: based on the Postscript/NeWS model.

The jury is still out, but diversification, poor decisions by Javasoft (using a *The Right Thing* philosophy) are turning a sure thing into a loss.

The Theory Applied to a Static Language

C++:

- Technological model:
  ✓ Old technology: Yes, Simula in Lab 20 years earlier, but obscure
  ✗ Tried before: not really—Simula not in US; Smalltalk, Actors in the lab
  ✗ Appeals to market need: not entirely—OO hype
  ✗ Smallest investment: not really, somewhat of a lengthy development process
  ✓ Minimize typing: yes, Cfront
- Design model:
  ✗ Simplicity: no, complex syntax, hybrid language, difficult semantics
  ✗ Minimal completeness: maybe not—too much coverage
  ✓ Minimal correctness: yes
  ✓ Minimal consistency: yes
  ✗ Minimal abstraction: maybe not
- Implementation:
  ✓ Fast: mostly
  ✓ Small: mostly
  ✓ Interoperate: yes, mostly
- Environment:
  ✗ Cultural change: no (tagged along at the tails of C and OOP)
  ✗ Quickly adapted: no, developed for particular purposes
  ✓ Gambler's Ruin-safe: yes, AT&T, Microsoft

The Theory Applied to a Static Language

C++:

- Language requirements:
  ✓ Runs everywhere: yes
  ✗ Minimal computer requirements: maybe not
  ✓ Simple performance model: mostly
  ✗ Minimal mathematical sophistication: maybe not
  ✓ Minimally acceptable for purpose: yes
  ✓ Similar to popular language: yes, but similarity disguises steep learning curve
  ✓ Gurus: yes

Verdict:

C++ seems to have most of the right stuff, but perhaps appeared at the wrong time (too early), and its obscure roots (Simula—not popular or even known in the US) negates a lot of the effect. Smalltalk was still in the labs at the point C++ came out. Most importantly, there doesn't seem to be an independent cultural or environmental change to hook onto. C++ was designed too smartly and too deliberately.

The surprise verdict is that maybe this language won't get beyond niche acceptance in the long run.

How the Theory Nullifies Business Misconceptions

- Business managers believe (falsely) that first-to-market wins
- Engineers believe (falsely) that inventiveness and best technology wins
- Marketing people believe (falsely) that market investment wins

But in a well-executed application of standard business practice in software:

- Management requirement for first-to-market and low investment creates unrealistic schedules and under-funded projects
- Engineers, believing themselves inventive, respond by:
  - selecting mature technology to avoid having to start over
  - cleverly piecing together a product from existing parts
  - leaving out unnecessary (and even some necessary) features
- Marketing people attach their programs to the environmental change—**which they believe they have created**—and (sometimes) increase awareness of the change and pave the way for the second, inexorably growth stage
- Therefore, the myths business holds conspire to execute the theory of technology acceptance

Why is Business Theory so Wedged?

- A mistaken belief that rationality dominates acceptance decisions and that rationality entails progress, but the following factors dominate:
  - risk aversion/safety
  - guessing the future
  - similarity to existing solutions
  - cost
- A mistaken belief that money=rationality
- A (possibly) mistaken belief that the marketplace is like the history of life
- A mistaken understanding of the basics of evolution
- The fortuitous situation that the real behavior of the marketplace and the true nature of technology adoption turn stupid decisions into good ones (two wrongs make a right)
- The need for solace in human and evolutionary progress