



15.965 Technology & Strategy

Technology and Strategy

Modularity, open interfaces, open standards and open source
Michael A M Davies

Strategy and technology

Demand opportunity

- segments
- behaviour change
- diffusion and adoption
- chasm

Technological infrastructure

- architecture
- parameters
- envelope and trade-offs
- innovation trajectory

Co-evolution & transitions

- episode, era
- dominant design
- radical vs incremental
- architectural vs modular
- disruptive

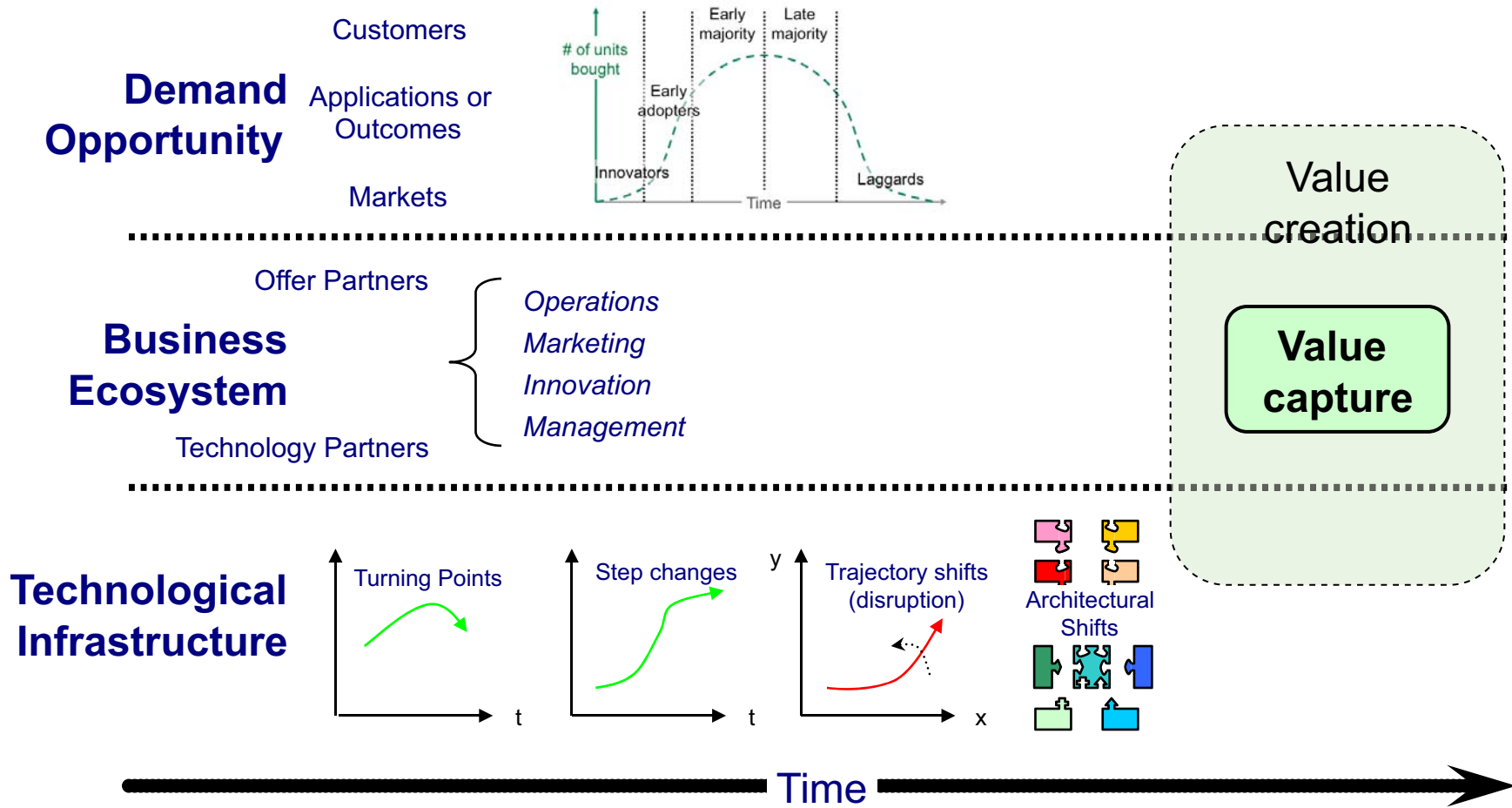
Business ecosystems, value creation and value capture

- niche
- lead/follow
- co-opetition
- inimitability
- focus on locus of value
- standards
- modularity

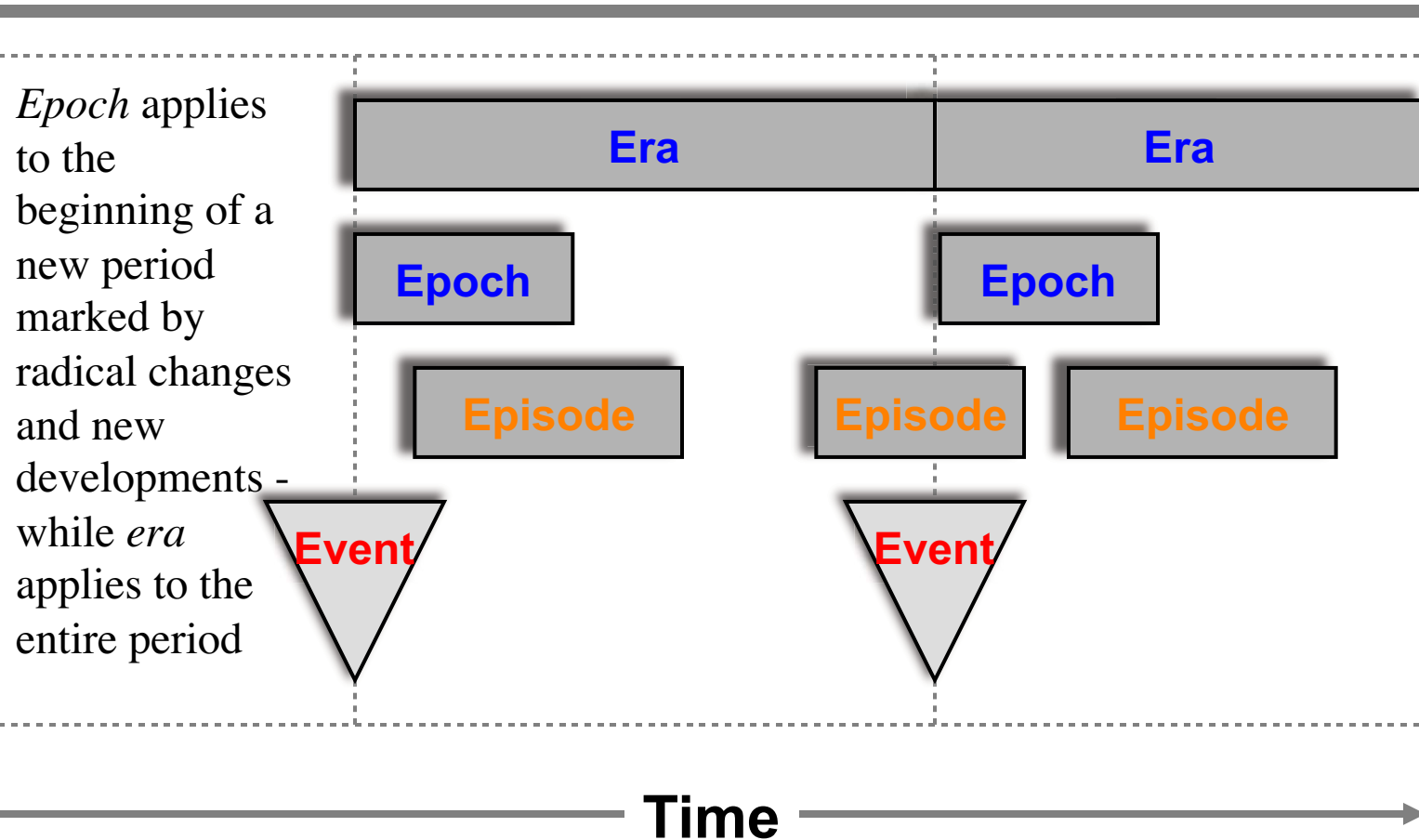
Decision making

- common goals, balanced power
- focus on timely, objective facts
- embrace uncertainty
- several options

Two simultaneous challenges: *create value*, at the same time ensure that you can *capture value*



Some terminology for timelines and transitions



Roget's New Millennium™ Thesaurus, First Edition (v 1.3.1) © 2007



Era

noun

1. a period of time marked by distinctive character, events and so on¹
2. the period of time to which anything belongs or is to be assigned¹
3. a period of time as reckoned from a specific date serving as the basis of its chronological system²

1: Random House Unabridged Dictionary, © Random House Inc. 2006

2: American Heritage® Dictionary, © 2000 Houghton Mifflin





Epoch

noun

1. a period of time marked by distinctive character, events and so on¹
2. the **beginning** of a distinctive period in the history of anything¹
3. a **point of time** distinguished by a particular event or state of affairs¹
4. a notable event that marks the **beginning** of a period of history, especially one considered remarkable or noteworthy²

1: Random House Unabridged Dictionary, © Random House Inc. 2006

2: American Heritage® Dictionary, © 2000 Houghton Mifflin





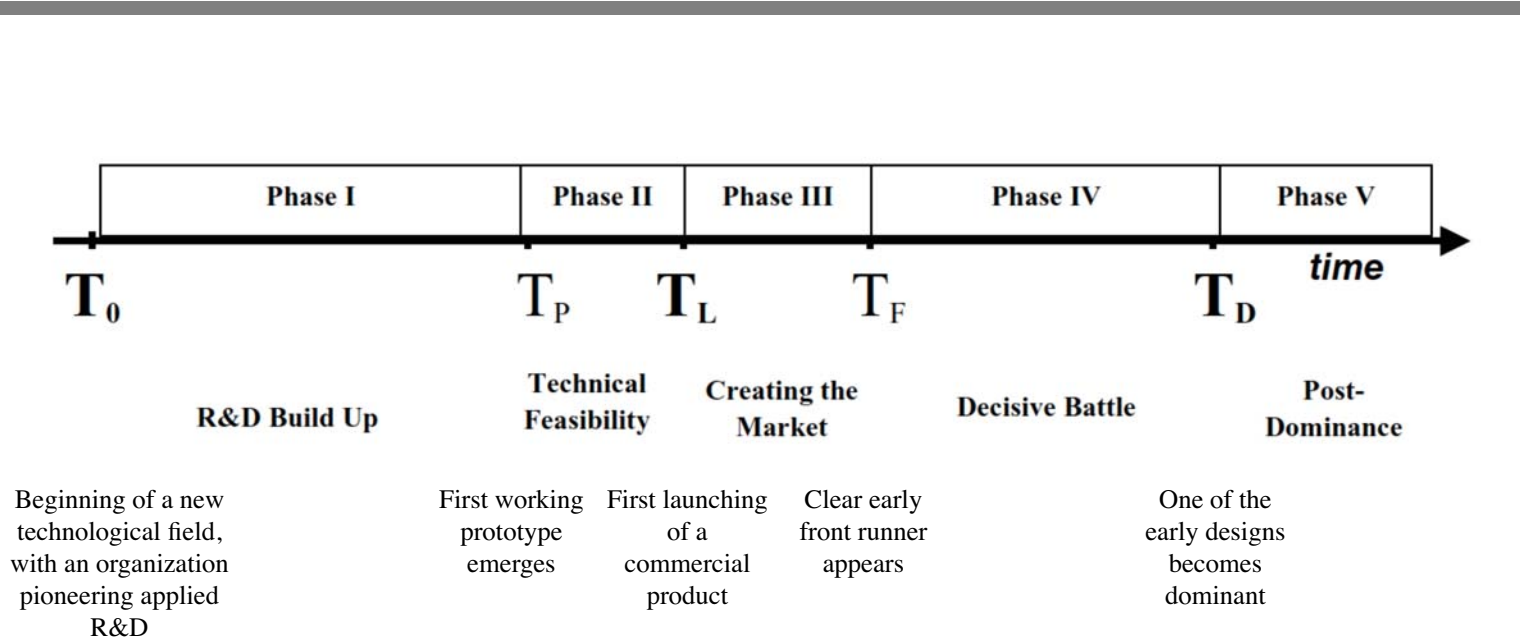
Episode

noun

1. a portion of a narrative that relates an event or a series of connected events and forms a coherent story in and of itself¹
2. an incident in the course of a series of events²

1: American Heritage® Dictionary, © 2000 Houghton Mifflin
2: Random House Unabridged Dictionary, © Random House Inc. 2006

There are distinct stages in the battle for technological dominance



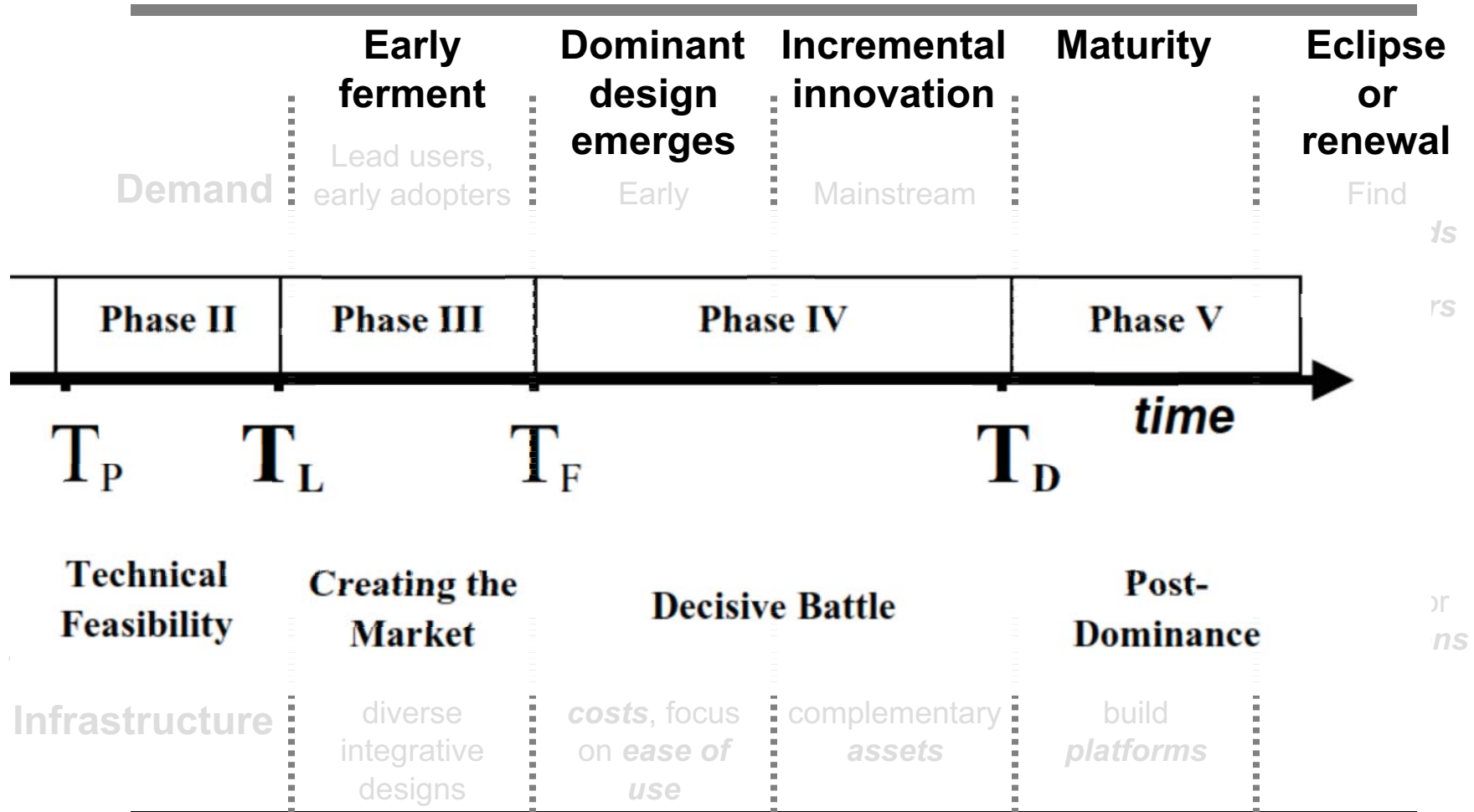
Courtesy Elsevier, Inc., <http://www.sciencedirect.com>. Used with permission.

Fernando Suarez, "Battles for technological dominance: an integrative framework", Research Policy, Volume 33, 2004, pages 271-286

These phenomena often result in characteristic {product | industry} ‘life-cycle’ (YMMV)

	Early ferment	Dominant design emerges	Incremental innovation	Maturity	Eclipse or renewal
Demand Opportunity	Lead users, early adopters - high payoff, low switching costs	Early mainstream - usability, cost more important	Mainstream customers - soft factors, aesthetics	Saturation, segmentation, customization	Find <i>new needs</i> or <i>new customers</i>
Business Ecosystem	Many entrants - diverse business models	Decisive battles for leadership	Intensifying competition, early consolidation	Fierce competition, consolidation around majors and minors	
Technological Infrastructure	Make it work - innovate on <i>performance</i> , diverse integrative designs	Select optimal architecture, drive down <i>costs</i> , focus on <i>ease of use</i>	Provide broader offer, rationalize <i>portfolio</i> , build complementary <i>assets</i>	Develop <i>broad portfolio</i> , build <i>platforms</i>	Search for <i>new options</i>

Reconciling these two models



GNU and Linux

- (Lead) user (from CSAIL at MIT) wanted **control**
- Developed key complements:
 - compiler and editor
 - license – GPL/“copyleft”
- Linus Torvalds: ‘91, <10k lines, Intel’s ‘386 architecture
- Increased functionality
- Wider variety of hardware platforms
- Network effects

- Increasing complexity

Red Hat

- (Lead) user finds Linux difficult to install and use
 - couldn't afford a Unix workstation
 - The Halloween Release
- Initially inferior, did not appeal to mainstream, potential for rapid innovation, surpasses performance
- Combine *build* with *distribution channel*
- Key innovations
 - installer
 - package manager
- Time-paced release – work back from ship date

Red Hat - update

- 2004 ranks first in value amongst CIOs
- Acquires AOL's Netscape **server** software
- Partners with IBM
- 2005, Enterprise Linux 4 wins OS and Server Product of the year at Techworld
- IPO in August 1999 – shares triple

- ~2,200 employees
- ~\$500 million in revenue, ~\$4,000 million market cap

Microsoft's take on Linux

- *OSS poses a direct, short-term revenue and platform **threat** to Microsoft, particularly in server space*
- *the intrinsic parallelism and free idea exchange in OSS has **benefits that are not replicable** with our current licensing model and therefore present a **long term developer mindshare threat**.*
- *...**commercial quality** can be achieved / exceeded by OSS projects*
- *OSS is long-term credible **FUD tactics can not... combat it**.*
- ***Linux outperforms many other UNIXes***
- *Linux can win as long as services / protocols are commodities.*
- ***OSS projects have been able to gain a foothold in many server applications because of the wide utility of highly commoditized, simple protocols. By extending these protocols and developing new protocols, we can deny OSS projects entry into the market.***
- *OSS evangelization scales with the size of the Internet much faster than our own evangelization efforts appear to scale*



Why does open source work (economically, not technically)?

Co-operate

⇒ *create value*

- OS infrastructure, not a basis for differentiation
- “...*all the users of that infrastructure have a reason to help you build it and ensure that it continues to develop.*”
- A modular system – enabling decoupled development
- Expose interfaces and inner workings of modules

Compete

⇒ *capture value*

- Deliver complements
- Package sources and patches
- Provide professional services
- Sell hardware that runs Linux



Modularity is becoming more prevalent, increasing value *creation*, challenging value *capture*

- Falling costs of co-ordination make modularity easier
- Modularity, where it can be employed effectively, can accelerate value creation
 - once dominant design established, hence stable architecture and modular interfaces
 - and ultimate performance is not critical
 - autonomous or modular innovations, in this context
- Very challenging for value capture
 - loss of control for leaders
 - rapid, diverse innovation
 - revenues and value widely dispersed



Modularity

noun

1. the use of individually distinct functional units, as in assembling an electronic or mechanical system¹
2. designed with standardized units or dimensions, as for easy assembly and repair or flexible arrangement and use²

1: Random House Unabridged Dictionary, © Random House, Inc. 2006

2: American Heritage® Dictionary, © 2000 Houghton Mifflin



Modularity → decoupling

- *“When a product or process is ‘modularized,’ the elements of its design are split up and assigned to modules according to a formal architecture or plan.”*
- *“From an engineering perspective, a modularization generally has three purposes:
 - to make complexity manageable
 - to enable parallel work
 - to accommodate future uncertainty”*

Modularity in the Design of Complex Engineering Systems,
Carliss Y. Baldwin and Kim B. Clark, HBS Working Paper, January 2004

Interfaces

Modular

- Customer understands and can specify key parameters
- Can be measured and tested reliably and unambiguously
- Understand how variation affects system performance
- Market can function effectively
- Codified knowledge
- Difficult to protect

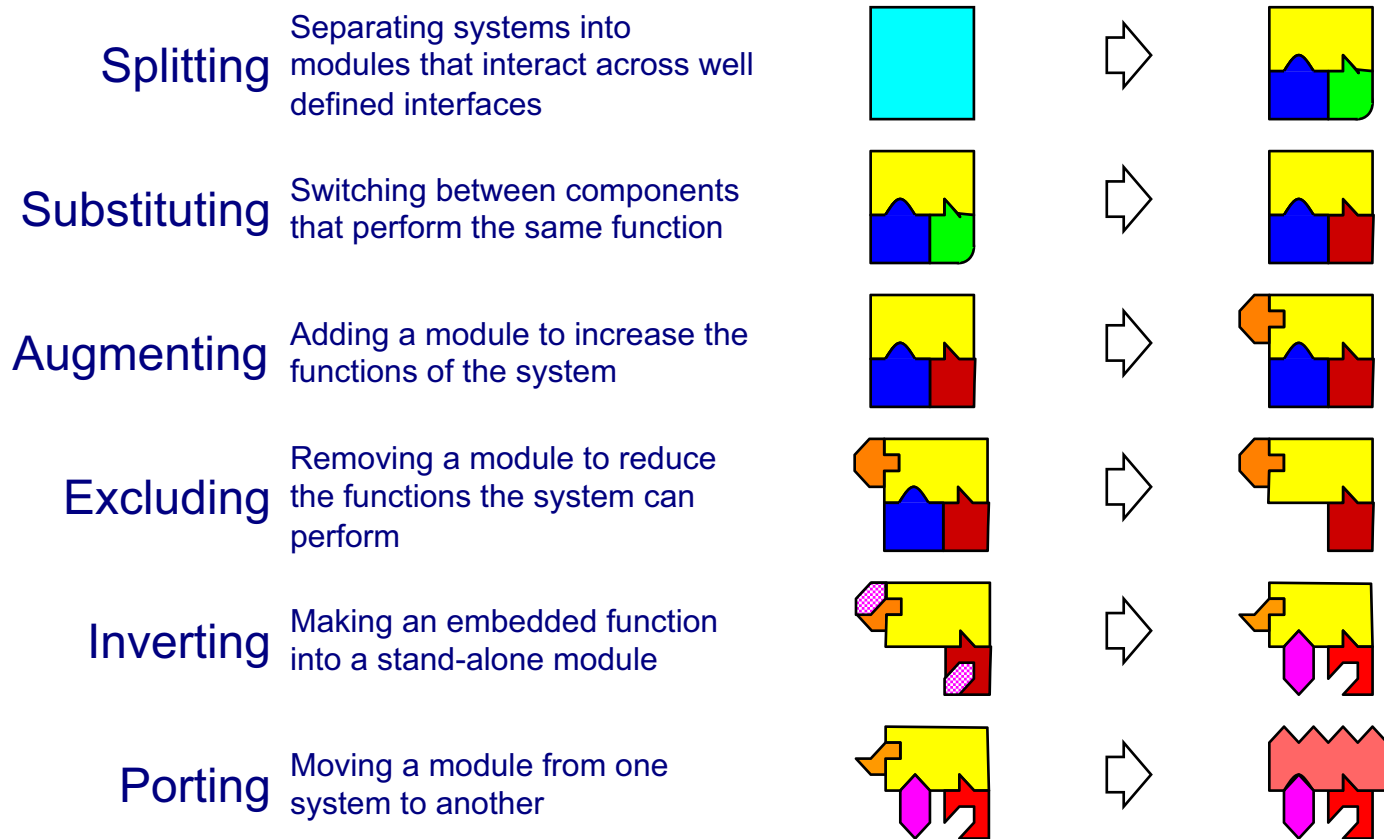
Interdependent/ Systemic/Integral

- Associated with optimizing design for ultimate performance
- Unstructured technical dialogue
- Necessary information for market does not exist
- Management and integration most efficient coordinating mechanisms

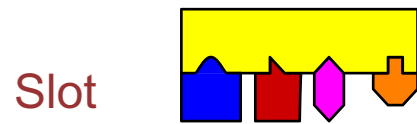
IBM System/360

- First modular computer, conceived as a family of computers
 - different sizes suitable for different applications
 - same instruction set
 - standard interfaces for peripherals
- Design rules and decentralized development
 - Central Processor Control Office defines rules
 - each team full control over hidden elements
- Wildly successful, drove other players out of the market
- BUT undermined IBM's dominance in the long run - through emergence of plug-compatible modules

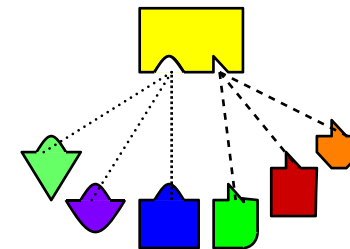
There are six modular operators that together enable a very wide range of system designs



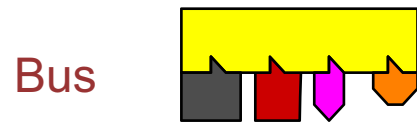
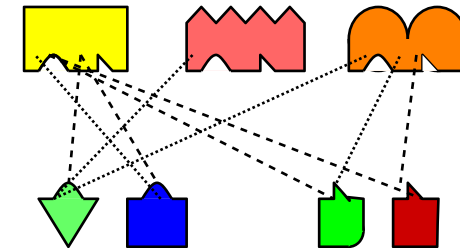
The resulting systems can exhibit several different types of modularity



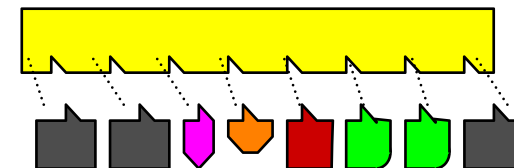
Component Swapping



Component Sharing



Bus



□

**Modular
platforms can
be a very
effective
vehicle for
diverse offers**

Image removed due to copyright restrictions.

Strategic options

Architect

- For system innovations, may require broad scope of activities at the outset
- Create design rules, define visible information
- Convince people this architecture will prevail
- As modularity established, lead the evolution of the business ecosystem

Module player

- Conform to the architecture, interfaces and test protocols established by others
- Master the hidden information involved
- Rely on superior execution



Linux 0.01

The architecture of Linux is significantly more modular than the architecture of Mozilla

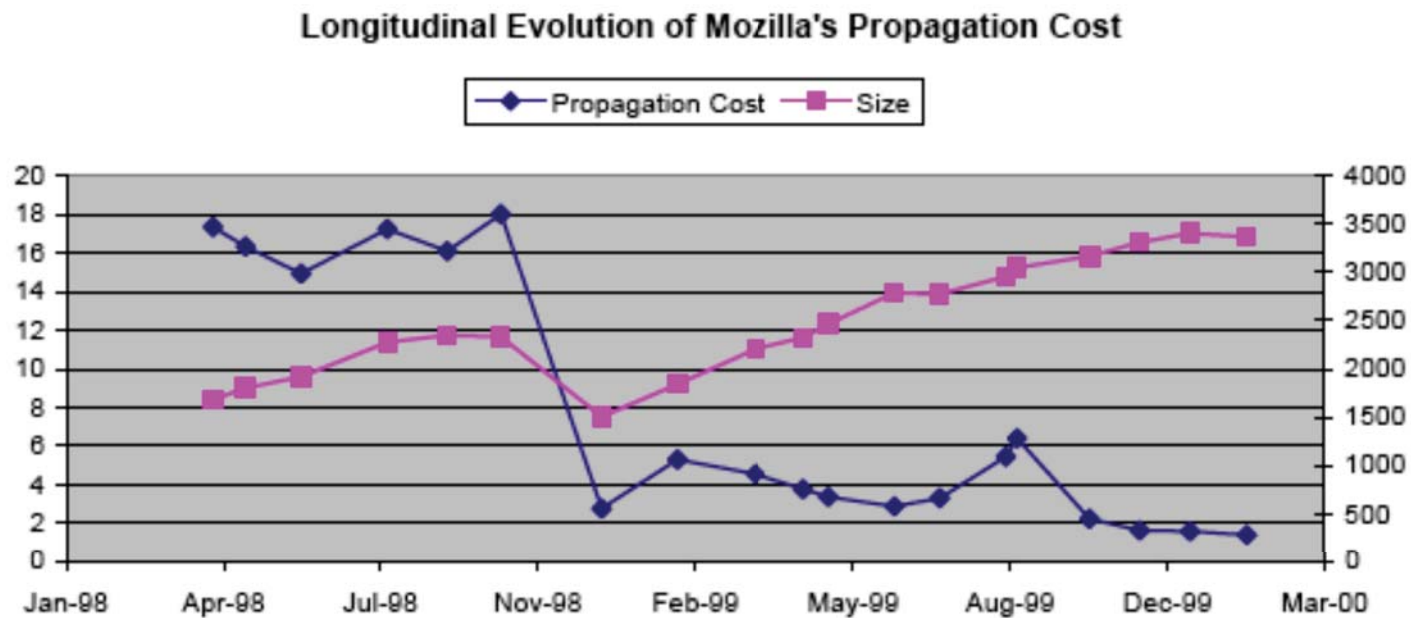
And than OpenSolaris

...or XNU

Mozilla – before and after a purposeful re-design effort – modularity to allow participation

MacCormack, Alan, John Rusnak, and Carliss Y. Baldwin. "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code." *Management Science* 52, no. 7 (July 2006)

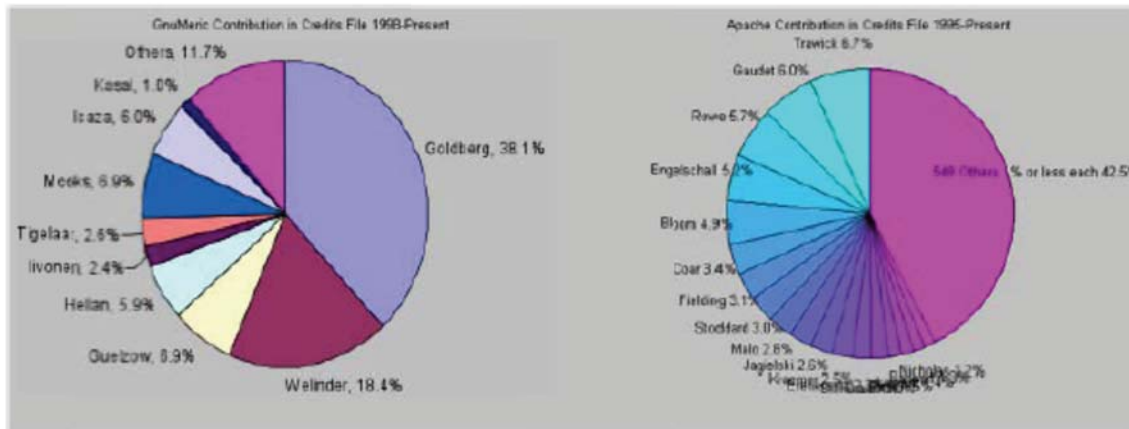
The payoff...





Open source code has much lower propagation cost than conventional proprietary approaches

Product Category	“Open”	“Closed”	Test Stat
1: Financial Mgmt	7.74%	56.06%	p<0.1%
2: Word Processing	8.25%	41.77%	p<0.1%
3: Spreadsheet	23.62%	54.31%	p<0.1%
4a: Operating System	7.18%	22.59%	p<0.1%
4b: Operating System	7.21%	24.83%	p<0.1%
5: Database	11.30%	43.23%	p<0.1%



Key to value capture is *focus on locus of value*

- “... *the virtuous virtuals have carefully nurtured and guarded the internal capabilities that provide the essential underpinnings of competitive advantage... they invest considerable resources to maintain and extend their core competences [because without them] their strategic position in the network would be short-lived*”

Henry Chesbrough and David Teece, “Organizing for Innovation: When is Virtual Virtuous?”,
Harvard Business Review, August 2002, pages

- “*Attractive profitability seems to flow ... to the point at which unsatisfied demand for functionality, and therefore technological interdependency exists.*”

Clayton Christensen and others, “Disruption, disintegration and the dissipation of differentiability”,
Industrial and Corporate Change, 2002, pages 955-993



Locus

noun

1. a center or focus of great activity or intense concentration¹
2. a center or source, as of activities or power²

1: American Heritage® Dictionary, © 2000 Houghton Mifflin
2: Random House Unabridged Dictionary, © Random House, Inc. 2006

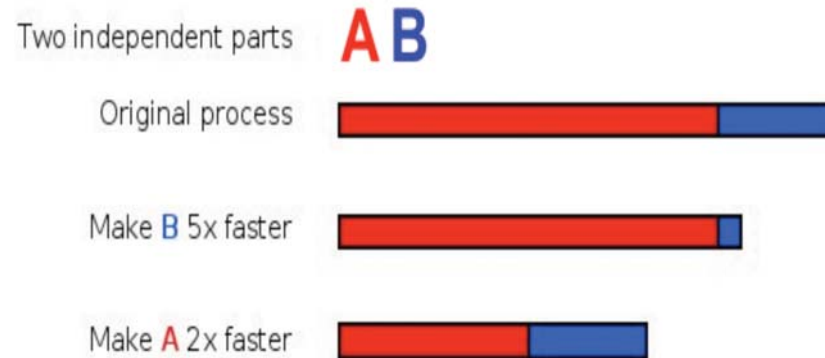


Amdahl's Law: "...make the common case fast..."

- Amdahl's Law is concerned with the speedup achievable
 - from an improvement to a computation
 - affects a proportion P of that computation
 - where the improvement has a speedup of S

- Amdahl's Law states that the overall speedup of applying the improvement will be

$$\frac{1}{(1 - P) + \frac{P}{S}}$$



“God grant me the serenity to accept the things I cannot change (much); courage to change the things I can (a lot); and wisdom to know the difference.”

- Reinhold Niebuhr



Sun Microsystems – intense focus on performance bottleneck

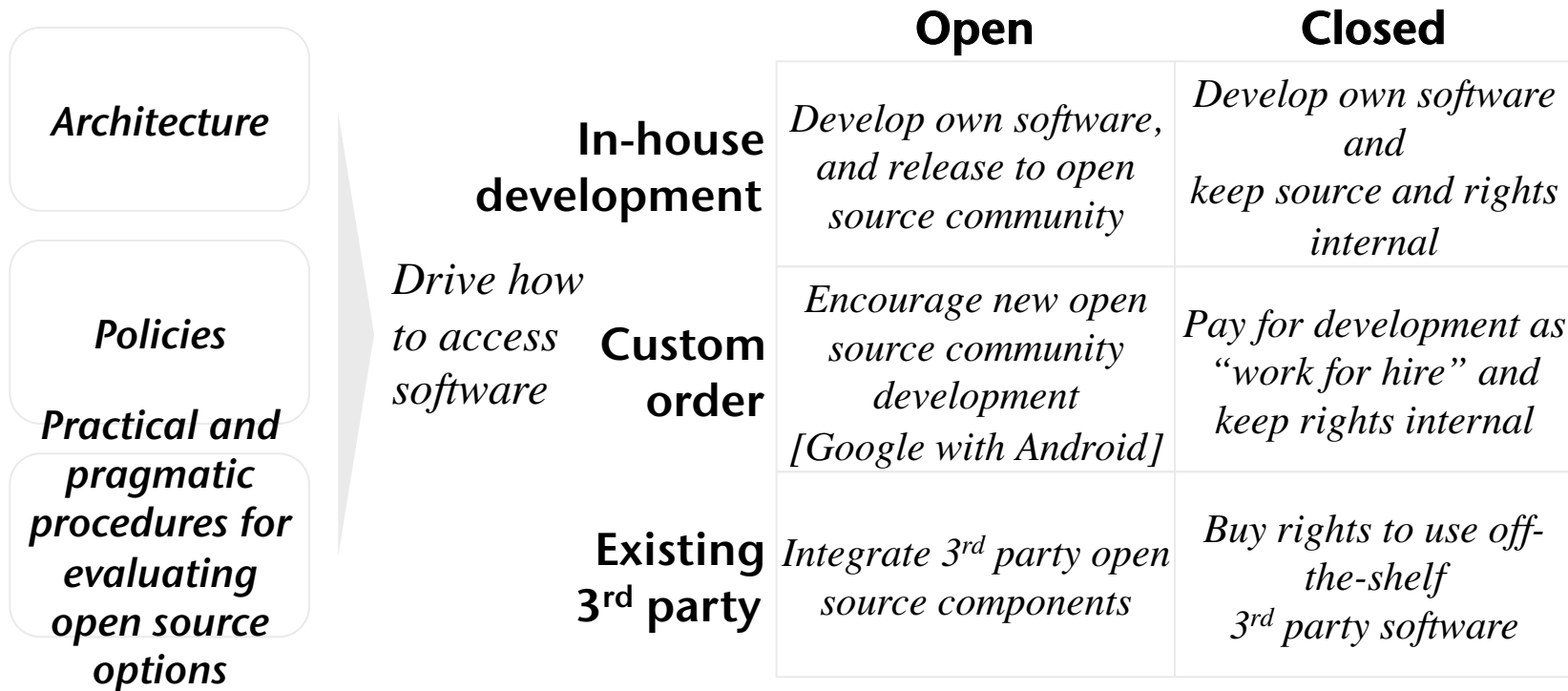
Image removed due to copyright restrictions.



Interestingly, Sun Microsystems has become a major open source advocate

- Made Java™ open source
- OpenSolaris
 - build a developer community
 - derived from Unix System V – only open version
- OpenSPARC
 - processor architecture design, application development
 - building community around it to advance it
- OpenStorage
 - built around ZFS
 - combine open source software with industry standard hardware

Leveraging open source requires a disciplined approach, from legal and strategic perspectives



Source: Olswang Open Source Summit, 9 Nov 2007; private discussion and communication with Nigel Swycher of Olswang, and Heather Meeker of Greeberg Taurig, and others



The key concern with open source is inherited obligations, “copyleft” licenses

Main kinds of open source license

<i>Principal form of License</i>	<i>Characteristics</i>
General Public License – GPL of the Free Software Foundation	“Copyleft” – most open; prohibits proprietary forks, strongly driving openness If you add code, and distribute software, you must grant unaltered GPL to combined program - Linux Kernel uses this (GPLv2)
Apache Software License (ASL)	Allows proprietary extensions or distributions Code issued with this license can be included on closed products Much of Android software uses this (Apache v2)
Berkeley Software Distribution (BSD Unix)	Few restrictions on what you can add and how you can limit openness of what you add Apple uses this



Resources

- <http://opensource.mit.edu>
- <http://www.opensourcestrategies.org/>

- <http://code.google.com/android/what-is-android.html>
- <http://www.osxbook.com/>
- <http://www.kernelthread.com/mac/osx/>
- <http://arstechnica.com/reviews/os/mac-os-x-10-5.ars>
- <http://www.roughlydrafted.com>*
- <http://opensolaris.org/os/>

MIT OpenCourseWare
<http://ocw.mit.edu>

15.965 Technology Strategy for System Design and Management
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.