Open Innovation Learning

Theory building on open sourcing while private sourcing

David Ing

Foreword by Jim Spohrer

http://openinnovationlearning.com

Book launch
February 21, 2018
OCAD University, Toronto

Image CC-BY-SA: Celina Laurette (2017) *Escaping from Plato's Cave*

Agenda

1. Introductions	Peter Jones
2. Outline	David Ing
3. Commentary	Tim Lloyd Stephen Perelgut
4. Q&A	Peter Jones
5. Reception	(signing postcards!)

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

a. Open Innovation Learning

b. Open Innovation Learning

c. *Descriptive* theory building

d. *Normative* theory building

e. *OIL* + Systems Thinking

Open Innovation (2003): ~~Xerox PARC~~; IBM √; Intel √, Lucent √

Contents

5

From Closed to Open Innovation

The Transformation of the IBM Corporation

THE VERY successful model of Closed Innovation within large firms has gradually given way to a more diffused, more externally focused way of organizing innovation. Younger companies and start-ups have eschewed the closed approach from the time of their founding, but the question remains whether or how an established company might move from a Closed Innovation mind-set to an Open Innovation mind-set.

The IBM Corporation has made such a transformation. Because of the company's long and storied history, the account here will necessarily be selective and organized around the themes of Closed Innovation and Open Innovation. IBM's transition was far from easy. In fact, it took a near-death experience to force it to make the shift. And many thousands of people had to be laid off along the way, so not everyone in the company was able to make this shift. Nonetheless, despite the layoffs and the write-offs, IBM's experience shows that even large, successful organizations from the days of Closed Innovation can (at some cost) become far more open in their approach to innovation. What's more, large companies can continue to profit from their innovation investments, albeit in different ways than they did before.

Closed Innovation Success at IBM: 1945–1980

It would be hard to overstate the impact that IBM has had on the computer industry. From the inception of computers during World War II up until 1980, IBM was a central player—in fact, *the* central player—in

93

A *closed* paradigm assumes internal control for innovation; *open* sees ideas outward/inward from/to the firm and market

FIGURE I-1

The Virtuous Circle

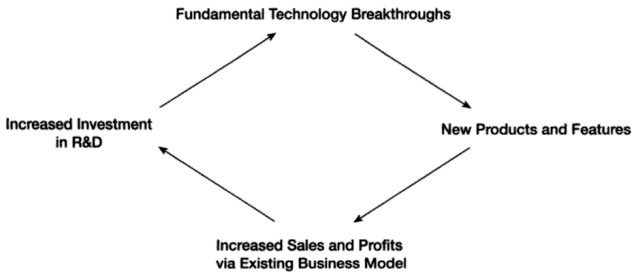


FIGURE I-2

The Closed Paradigm for Managing Industrial R&D

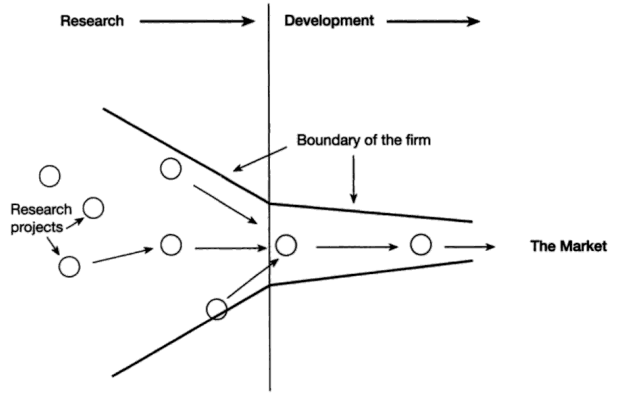
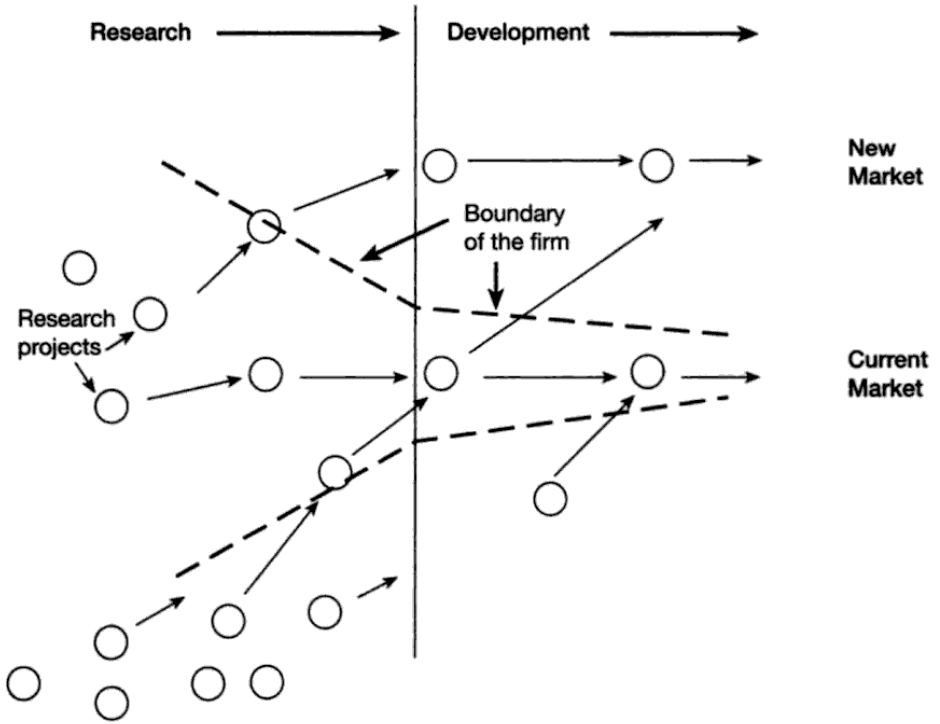


FIGURE I-4

The Open Innovation Paradigm for Managing Industrial R&D



Open Business Models (2006): open source development → success

was a reliable ally, at least in the short term. Collabra achieved some temporary protection of its ideas with Microsoft because it knew Microsoft would benefit from working with the smaller company, not because of the strength of Collabra's IP. And Collabra always maintained an outside option by courting other potential strategic partners—whether those other partners were Novell, IBM, or Netscape—if Microsoft didn't respect its IP.

MANAGING OPEN INNOVATION: IT TAKES AN OPEN BUSINESS MODEL

Whether you are in a large organization or a small one, chances are you need to open up your innovation processes. But to do this effectively, you must connect your business model to your innovation process. Companies that are large typically enjoy strong business models. However, they face challenges and risks that small companies do not. It is hard for large companies with successful business models to change them to exploit Open Innovation opportunities. Small companies, on the other hand, lack the strong business model and resources to enable them to exploit the opportunities of Open Innovation without fear of being copied by a larger foe. IP protection is one of the many tools needed in their business model to achieve success.

In the next chapter, we will look more carefully at the external environment surrounding the company's business model. Before doing that, however, we must consider a frequent objection that is raised about the idea of managing Open Innovation in the context of your business model: what about open source software development?

OPEN SOURCE: A SUCCESSFUL TECHNOLOGY WITHOUT AN APPARENT BUSINESS MODEL

One of the central tenets of my previous book, *Open Innovation*, is that business models are essential to unlocking latent value from a

technology. In the introduction, the book asserts: "There is no inherent value in a technology per se. The value is determined instead by the business model used to bring it to market. The same technology taken to market through two different business models will yield different amounts of value. An inferior technology with a better business model will often trump a better technology commercialized through an inferior business model."

Open source software development seems to challenge this claim. By construction, open source software is created without any one firm owning the ability to exclude others from using technology, provided that these other firms observe the open source requirements. Enhancements to the code are available to everyone on an equal basis.

Is open source's success simply an exception to the general rule, is this success due to a business model of a different kind, or is there something fundamentally wrong with the claims of *Open Innovation* regarding the importance of business models? What I hope to show now is that the evolution of the open source software movement is being propelled by the emergence of clear and distinct business models built around it. I think of them as "open source business models."

Open Source Software: A (Very) Brief Introduction

There have been numerous studies of the open source software community, ranging from enthusiastic proclamations of its benefits (e.g., Eric Raymond, *The Cathedral and the Bazaar*), to condemnations of it as pernicious to innovation (such as the comments by Steve Ballmer, CEO of Microsoft, likening Linux to a "cancer").[22] More scholarly examinations of the open source software approach can be found in "The Simple Economics of Open Source," "Guarding the Commons," and "How Open Is Open Enough?"[23] Listings of open source research resources are abundant online.[24]

And this wealth of study takes no account of the enormous online literature that discusses many elements of open source. This ranges from blogs, to communal places like Slashdot (www.slashdot.org), to repositories of open source software code, such as SourceForge.net.

As a result, we now know a great deal about how open source software development works. It is a collaborative, community model

of development, based on a process that does not allow any contributor to exert a proprietary claim to intellectual property on any portion of the code being developed within the open source framework. (However, the technical legal status of open source software is actually varied and complex, as different projects employ different licensing arrangements, which vary in the rights those arrangements convey to developers to use their contributions in other, proprietary software.)

What You Don't Read About: Open Source Business Models

One doesn't read much about business models in open source software development. There are strong social norms and legal protections that have been crafted to discourage people from profiteering on the work of their peers. There are even frequent postings on well-frequented Web sites that identify cases where the norms of the group appear to be violated (though there have been few, if any, legal sanctions against violators).

However, occasional crises reveal parties that have developed business models to profit from the adoption of open source software. One crisis of this kind was the threat by the Santa Cruz Operation (SCO) to enforce its alleged IP rights (contained in a version of Unix it purchased from Novell) over the code being widely circulated in the Linux community. It subsequently sued IBM for $1 billion for allegedly devaluing SCO's Unix license. Separately, SCO sued users AutoZone and Daimler Chrysler for using Linux without a license to SCO's Unix.[25] While the open source community was very upset over these lawsuits, a very different response came from a group of companies that included Intel, IBM, Hewlett-Packard, Novell, and Red Hat. These companies banded together to pool resources into a fund to indemnify customers of open source software for legal expenses they might incur in defending themselves from a lawsuit, should they choose to use open source software. Separately, IBM appears to have taken the lead role in defending against the SCO suit. Without demeaning IBM's sincerity in any way, committing substantial resources in this manner is a highly reliable indicator that IBM's business model, and the business model of the other companies just mentioned, ben-

Joy's law (1995) learned from open specifications projects at Sun

... the smartest people
in the world
don't all work for us.

The trick is to make it
worthwhile for
the great people
outside your company
to support
your technology.

Most of them work for
someone else.

Innovation moves faster
when the people elsewhere
are working
on the problem with you.

Source: Bill Joy, interviewed by Brett Schlender and Michael H Martin (1995). "Whose Internet Is It, Anyway?", *Fortune* v132.n12 (1995), pp. 120–142.

Laws of copyright in the *Berne Convention* (1886) were amended by *The Open Source Definition* (1999) and *Creative Commons* (2002)

WIPO Tools ↓

Home › Knowledge › WIPO Lex › Treaty Secretariat

Berne Convention for the Protection of Literary and Artistic Works

of September 9, 1886,
completed at PARIS on May 4, 1896,
revised at BERLIN on November 13, 1908,
completed at BERNE on March 20, 1914,
revised at ROME on June 2, 1928,
at BRUSSELS on June 26, 1948,
at STOCKHOLM on July 14, 1967,
and at PARIS on July 24, 1971,
and amended on September 28, 1979

TABLE OF CONTENTS [1]

Open Source Initiative

The Open Source Definition

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

creative commons

search powered by Google

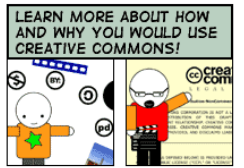
home | learn more | get content | discuss | technology | choose license

about us
faq
artists corners
licenses explained
examples
baseline rights
legal concepts
icommons
sampling
founders' copyright
science commons
weblog
press
press releases
store
donate

licenses explained

Choosing a License

Offering your work under a Creative Commons license does not mean giving up your copyright. It means offering *some* of your rights to any taker, and only on certain conditions.



view our comics:
A Spectrum of Rights
How It Works

What conditions? Our site will let you mix and match such conditions from the list of options below. There are a total of eleven Creative Commons licenses to choose from.

Attribution. You let others copy, distribute, display, and perform your copyrighted work – and derivative works based upon it – but only if they give you credit.

Noncommercial. You let others copy, distribute, display, and perform your work – and derivative works based upon it – but for noncommercial purposes only.

No Derivative Works. You let others copy, distribute, display, and perform only verbatim copies of your work, not derivative works based upon it.

Share Alike. You allow others to distribute derivative works only under a license identical to the license that governs your work.

Note: A license cannot feature both the Share Alike and No Derivative Works options. The Share Alike requirement applies only to derivative works.

Open sourcing is a behaviour (c.f. open source licensing, a legality)

Open sourcing
is a norm where
the resources of
system internals,
e.g. artifacts and
practices,
are **shared** in a
community
beyond the originators.

Simile: **Capturing Wild Fish**
[sharing naturally wild fish harvests]

Image: "Salmon Jumping Falls", CC-BY 2014 Katmai National Park and Preserve at https://www.flickr.com/photos/katmainps/16333875678/

Private sourcing is a behaviour counterposed to open sourcing

Private sourcing is coined as a norm where the resources of **system internals** are **reserved** within a privileged group.

Protections:

- Trade secrets (non-disclosure, non-compete)
- Patents (design with novelty, usefulness and non-obviousness)

Simile: **Aquafarming** [cultivating species in pens]

Image: "Aquaculture" CC-BY 2010 Burt Lum at https://www.flickr.com/photos/bytemarks/5211291608/

Behaviours seen: (i) open sourcing → private sourcing; (ii) private sourcing → open sourcing; (iii) open sourcing while private sourcing

(i) open sourcing enclosing to private sourcing

(ii) private sourcing disclosing to open sourcing

(iii) open sourcing *while* private sourcing

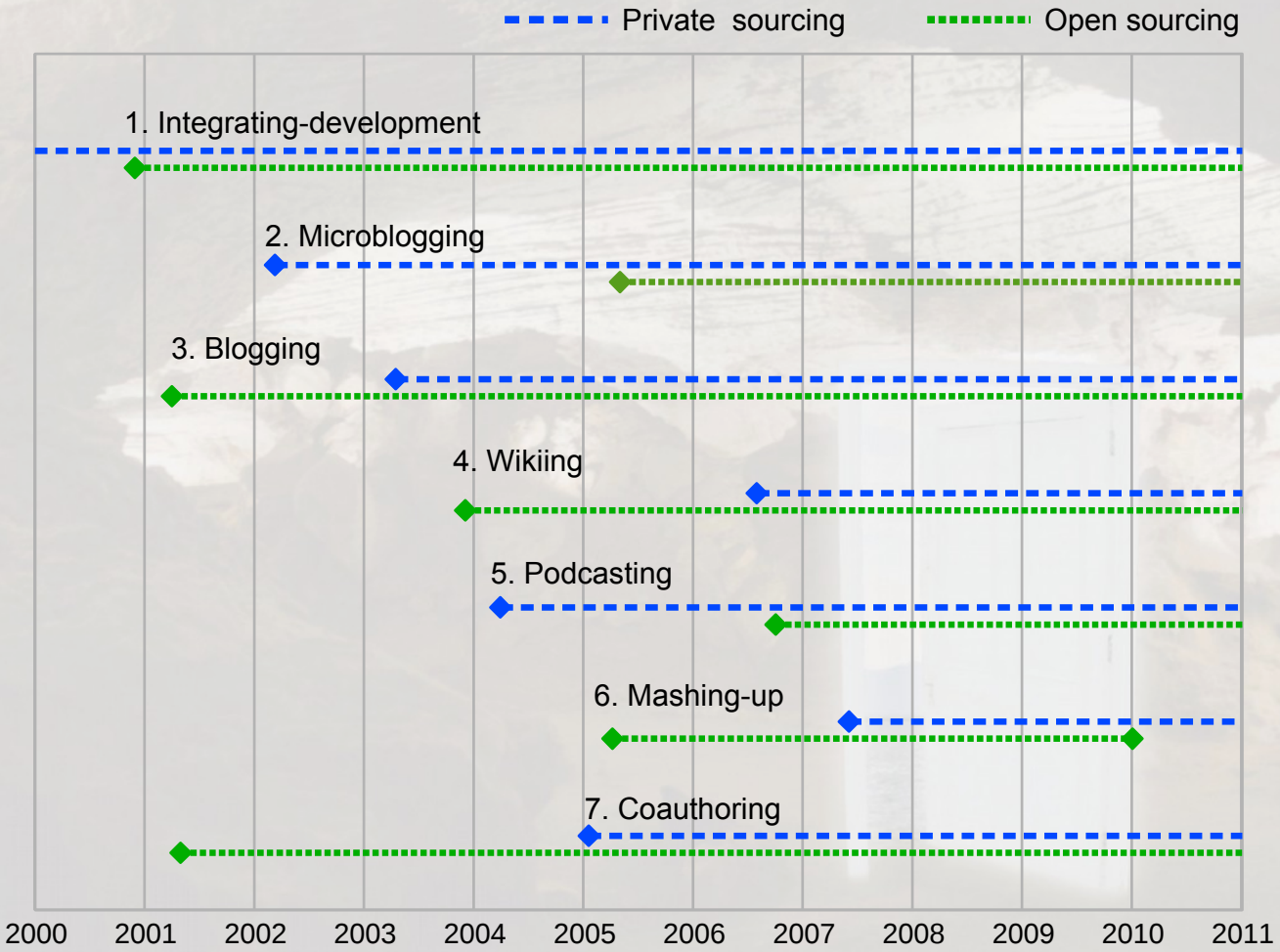
Simile:

Ocean Ranching

[conforming with aquacultural regulations + the law of the sea]

Image: “Aquaculture” CC-BY 2011 NOAA's National Ocean Service at https://www.flickr.com/photos/usoceangov/15961903877/

Open sourcing while private sourcing was learned at IBM, 2001-2011



Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

a. Open Innovation Learning

b. Open Innovation Learning

c. *Descriptive* theory building

d. *Normative* theory building

e. *OIL* + Systems Thinking

Organizational learning (1991): exploration+exploitation, communities of practice

ORGANIZATION SCIENCE
Vol. 2, No. 1, February 1991
Printed in U.S.A.

EXPLORATION AND EXPLOITATION IN ORGANIZATIONAL LEARNING*

JAMES G. MARCH

Graduate School of Business, Stanford University, Stanford, California 94305

This paper considers the relation between the exploration of new possibilities and the exploitation of old certainties in organizational learning. It examines some complications in allocating resources between the two, particularly those introduced by the distribution of costs and benefits across time and space, and the effects of ecological interaction. Two general situations involving the development and use of knowledge in organizations are modeled. The first is the case of mutual learning between members of an organization and an organizational code. The second is the case of learning and competitive advantage in competition for primacy. The paper develops an argument that adaptive processes, by refining exploitation more rapidly than exploration, are likely to become effective in the short run but self-destructive in the long run. The possibility that certain common organizational practices ameliorate that tendency is assessed.
(ORGANIZATIONAL LEARNING: RISK TAKING; KNOWLEDGE AND COMPETITIVE ADVANTAGE)

A central concern of studies of adaptive processes is the relation between the exploration of new possibilities and the exploitation of old certainties (Schumpeter 1934; Holland 1975; Kuran 1988). Exploration includes things captured by terms such as search, variation, risk taking, experimentation, play, flexibility, discovery, innovation. Exploitation includes such things as refinement, choice, production, efficiency, selection, implementation, execution. Adaptive systems that engage in exploration to the exclusion of exploitation are likely to find that they suffer the costs of experimentation without gaining many of its benefits. They exhibit too many undeveloped new ideas and too little distinctive competence. Conversely, systems that engage in exploitation to the exclusion of exploration are likely to find themselves trapped in suboptimal stable equilibria. As a result, maintaining an appropriate balance between exploration and exploitation is a primary factor in system survival and prosperity.

This paper considers some aspects of such problems in the context of organizations. Both exploration and exploitation are essential for organizations, but they compete for scarce resources. As a result, organizations make explicit and implicit choices between the two. The explicit choices are found in calculated decisions about alternative investments and competitive strategies. The implicit choices are buried in many features of organizational forms and customs, for example, in organizational procedures for accumulating and reducing slack, in search rules and practices, in the ways in which targets are set and changed, and in incentive systems. Understanding the choices and improving the balance between exploration and exploitation are complicated by the fact that returns from the two options vary not only with respect to their expected values, but also with respect to their variability, their timing, and their distribution within and beyond the organization. Processes for allocating resources between them, therefore, embody intertemporal, interinstitutional, and interpersonal comparisons, as well as risk preferences. The difficulties involved in making

*Accepted by Lee S. Sproull and Michael D. Cohen; received August 18, 1989.

1047-7039/91/0201/0071/$01.25

ORGANIZATION SCIENCE
Vol. 2, No. 1, February 1991
Printed in U.S.A.

ORGANIZATIONAL LEARNING AND COMMUNITIES-OF-PRACTICE: TOWARD A UNIFIED VIEW OF WORKING, LEARNING, AND INNOVATION*

JOHN SEELY BROWN AND PAUL DUGUID

Xerox Palo Alto Research Center and Institute for Research on Learning, 2550 Hanover Street, Palo Alto, California 94304
Institute for Research on Learning, 2550 Hanover Street, Palo Alto, California 94304

Recent ethnographic studies of workplace practices indicate that the ways people actually work usually differ fundamentally from the ways organizations describe that work in manuals, training programs, organizational charts, and job descriptions. Nevertheless, organizations tend to rely on the latter in their attempts to understand and improve work practice. We examine one such study. We then relate its conclusions to compatible investigations of learning and of innovation to argue that conventional descriptions of jobs mask not only the ways people work, but also significant learning and innovation generated in the informal communities-of-practice in which they work. By reassessing work, learning, and innovation in the context of actual communities and actual practices, we suggest that the connections between these three become apparent. With a unified view of working, learning, and innovating, it should be possible to reconceive of and redesign organizations to improve all three.
(LEARNING; INNOVATION; GROUPS; DOWNSKILLING; ORGANIZATIONAL CULTURES NONCANONICAL PRACTICE)

Introduction

Working, learning, and innovating are closely related forms of human activity that are conventionally thought to conflict with each other. Work practice is generally viewed as conservative and resistant to change; learning is generally viewed as distinct from working and problematic in the face of change; and innovation is generally viewed as the disruptive but necessary imposition of change on the other two. To see that working, learning, and innovating are interrelated and compatible and thus potentially complementary, not conflicting forces requires a distinct conceptual shift. By bringing together recent research into working, learning, and innovating, we attempt to indicate the nature and explore the significance of such a shift.

The source of the oppositions perceived between working, learning, and innovating lies primarily in the gulf between precepts and practice. Formal descriptions of work (e.g., "office procedures") and of learning (e.g., "subject matter") are abstracted from actual practice. They inevitably and intentionally omit the details. In a society that attaches particular value to "abstract knowledge," the details of practice have come to be seen as nonessential, unimportant, and easily developed once the relevant abstractions have been grasped. Thus education, training, and technology design generally focus on abstract representations to the detriment, if not exclusion of actual practice. We, by contrast, suggest that practice is central to understanding work. Abstractions *detached from practice* distort or obscure intricacies of that practice. Without a clear understanding of those intricacies and the role they play, the practice itself cannot be well understood, engendered (through training), or enhanced (through innovation).

*Accepted by Lee S. Sproull and Michael D. Cohen.

1047-7039/91/0201/0040/$01.25

In 1993, a new CEO declared a $8.9 billion restructuring charge for 2Q

The Newspaper of Information Systems Management
August 2, 1993, Vol. 27, No. 31, 100 Pages, $6/Copy, $48/Year

COMPUTERWORLD

Red ink soaks vendors

Gerstner bites bullet: Margins, write-offs contribute to $8B loss

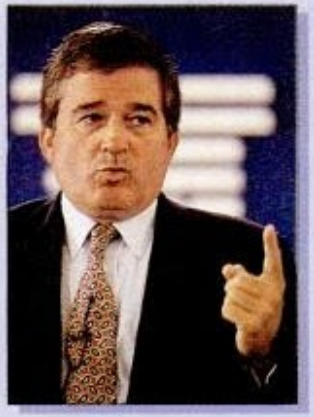
By Johanna Ambrosio
ARMONK, N.Y.

THE FIRST 100 DAYS

IBM Chairman Louis V. Gerstner's disclosure last week that 85,000 people will leave IBM by the end of next year — to the tune of $8.9 billion in write-offs — was the latest bullet-biting example of why he is winning good reviews after four months on the job.

Among the positives: his continuing focus on cutting operating costs, as underscored by last week's announcements (see story page 14); his increased emphasis on customers, as evidenced by the two or three he meets with each day; his appointment of strong outsiders to key management positions; and his decision to keep IBM as one large company with many discrete pieces, instead of breaking up the monolith into baby Blues.

IBM's changing shade of blue



Louis V. Gerstner has been busily injecting new blood into IBM. Outside appointments have included David Kallis, RJR Nabisco; Jerome York, Chrysler Corp.; Gerald Czarnecki, BankAmerica Corp.; and Abby Kohnstamm, American Express. Gerstner also appointed IBM insider Bernard Puckett as strategy guru.

HEAD COUNT

1993
255,000
(projected year-end total)

1992
301,000
(year-end total)

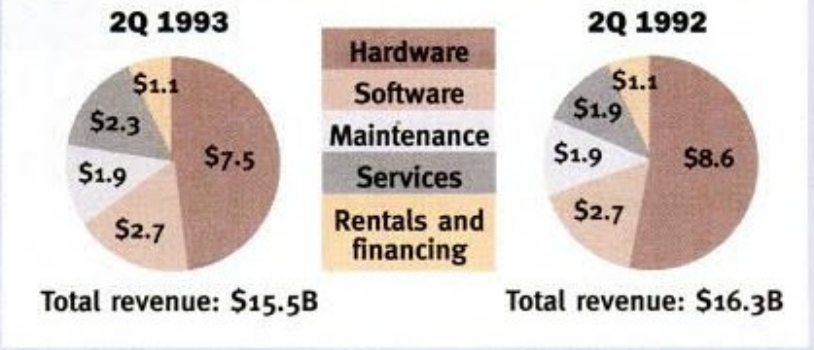
NET INCOME

2Q 1993
$(8 billion)*

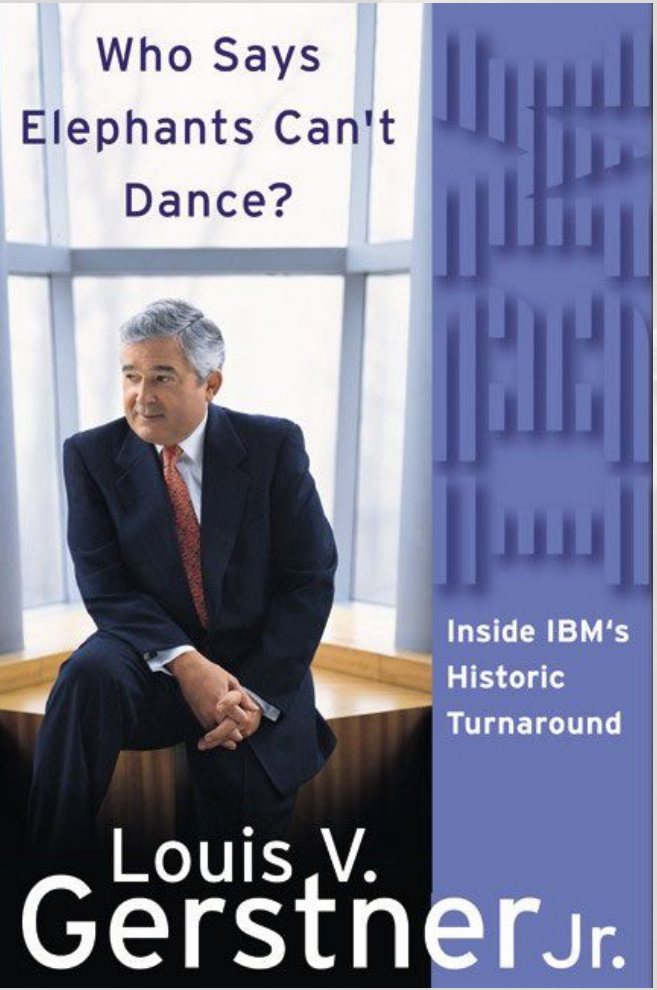
2Q 1992
$734 million

*Includes $8.9 billion restructuring charge

REVENUE (in billions)



The CEO found 300,000 IBMers “hiding” assets “owned” by internal units



In IBM's culture of “no” – a multiphased conflict in which units competed with one another, **hid things from one another**, and wanted to control access to their territory from other IBMers – the foot soldiers were IBM staff people. Instead of facilitating coordination, they manned the barricades and protected the borders.

For example, huge staffs spent countless hours **debating and managing transfer pricing terms between IBM units** instead of facilitating a seamless transfer of products to customers. Staff units were duplicated at every level of the organization because no managers trusted any cross-unit colleagues to carry out the work. Meetings to decide issues that cut across units were attended by throngs of people, because everyone needed to be present to protect his or her turf.

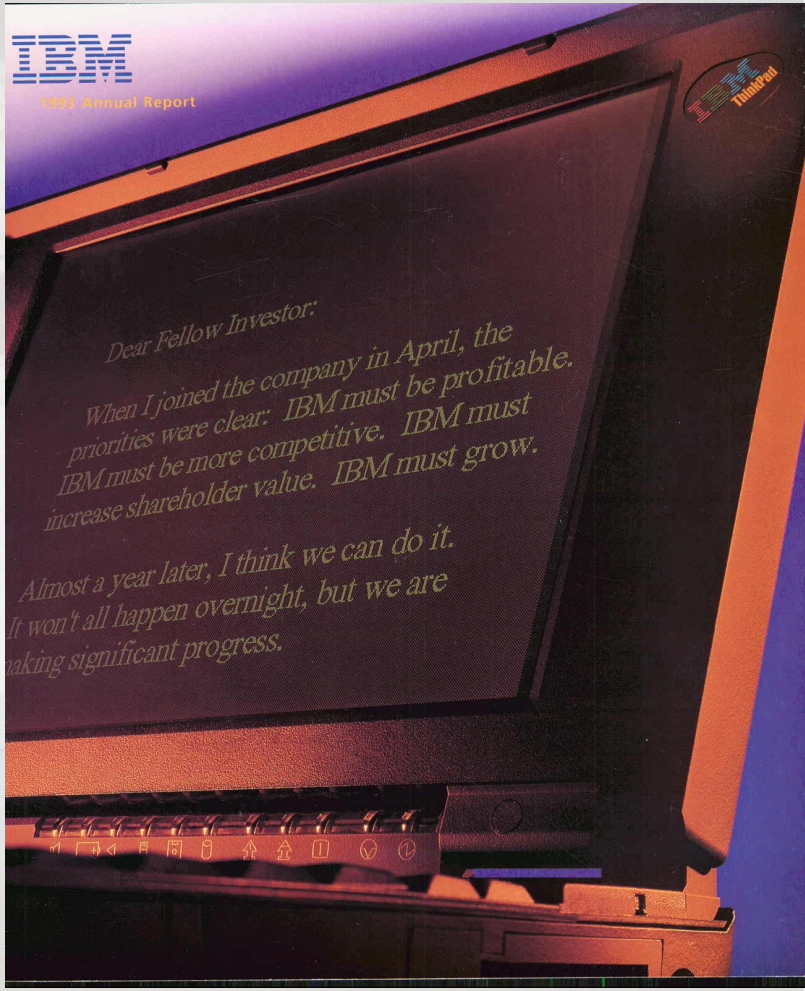
The net result of all of this jockeying for position was a very powerful bureaucracy working at all levels of the company – tens of thousands trying to protect the prerogatives, resources, and profits of their units; and thousands more **trying to bestow order and standards on the mob** (Gerstner, 2002, pp. 195–196).

Open sourcing for IBM was seeded in May 1993, at Chantilly

At a 200-customer forum on May 19 in Chantilly, VA, Gerstner said:
"One of the most important things I can say to you is there is **now a customer running IBM**".

CEO Gerstner "laid out" expectations:

- We would **redefine IBM** and its priorities starting with the customer.
- We would give our laboratories free rein and deliver **open, distributed, user-based solutions**.
- We would recommit to quality, **be easier to work with**, and reestablish a leadership position (but not the old dominance) in the industry.
- Everything at IBM would **begin with listening to our customers** and delivering the performance they expected.



Judith Dobrrzynski, "Rethinking IBM", Business Week, Oct. 4, 1993
http://www.judithdobrzynski.com/11233/cover-story-rethinking-ibm
Louis V. Gerstner (2002), *Who says elephants can't dance?* p. 48

In 1998, IBM funded *Apache Foundation* + invested in *WebSphere*

COMPUTERWORLD

The World's Technology Newspaper
News updates, features, forums: www.computerworld.com
June 29, 1998 • Vol. 32 • No. 26 • 124 pages • $4/Copy

IBM adopts freeware Apache Web server

By Carol Sliwa

IBM's ENDORSEMENT of the freely available Apache Group Web server — the most popular on the Internet — will provide an interesting option for corporations reluctant to use informally supported software.

Industry observers view the IBM plan as a smart move. But no one expects defections from the legions of corporate users who have already installed Netscape Communications Corp. and Microsoft Corp. World Wide Web server products — despite the allure of owning Apache's server source code.

"We have a fairly large investment in Netscape — not only code but the expertise," said John Swartzendruber, manager of enterprise architecture at Eli Lilly & Co. in Indianapolis. "The fact that IBM supports Apache probably wouldn't tip the balance in Apache's favor."

"I think it's great, as long as IBM doesn't take control of Apache," said Edward Bianco, chief information officer at Lowell General Hospital in Lowell, Mass. "But right now, I'm a big Microsoft shop, so it's hard for me to use it."

Eli Lilly's John Swartzendruber says his firm has a "large investment in Netscape"

Because Microsoft's Internet Information Server comes bundled free with its Windows NT operating system, it has been the easiest option for many customers. But IBM customers may be more inclined to consider the Apache server.

For instance, McDonald's Corp. in Oak Brook, Ill., has been using Lotus Development Corp.'s Domino server, but the company plans to check out other products as it rethinks its "stick with one platform" strategy. "Apache could be in the running," said Aaron Wiltz, a technical analyst at McDonald's. "It'll be more seriously looked at because IBM will definitely be putting the support that they put behind all of their products."

IBM is making both Apache and its Domino Go Web servers available as part of its new Web-Sphere Application Server, but the product also will support Netscape and Microsoft Web servers.

IBM plans to offer support for the Apache server only for customers who purchase the WebSphere product, said IBM program director Nigel Beck. But, if the market dictates that IBM provide support for the Apache server to non-WebSphere customers, the company might consider it, Beck said.

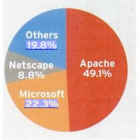
Of course, customers could buy the WebSphere Application Server for $795 to get support for Apache, even if they don't use WebSphere.

IBM plans to participate in the Apache Project in the same way other developers do: redistributing bug fixes and improvements to the informal group of programmers who developed and maintain the Apache software, Beck said.

IBM won't produce a "unique IBM version of Apache," Beck added, although it plans to add security features that Apache is prohibited from adding because of export controls.

Other companies have made similar additions, including C2Net Software, Inc. in Oakland, Calif., and Covalent Technologies, Inc. in Lincoln, Neb.□

Internet server market share



Base: June 1998 survey of 2.05 million distinct Web addresses

Source: Netcraft Web Server survey (www.netcraft.co.uk)

IBM's blessing of Apache bodes well for corporate users. See editorial, page 32.

OPINION

Big Blue blessing A little cheer went up in my heart when IBM recently adopted the free Apache Web server software to bundle with its own WebSphere line. This is the Big Blue blessing laid upon a truly open standard, on source code born and raised in the spirit of the Internet community. No marketing blitzkrieg. No empty talk about innovation. Just quiet, constant improvement in code quality — by volunteers.

About half the Web sites out there today use Apache freeware as their HTTP server, a critical component of Web applications that communicates with the browser. It's actually a very straightforward piece of software. It sits on the Internet and feeds Web pages to browsers as requested. No rocket science required.

But years of fine-tuning and bug-fixing by scores of Internet contributors produced a robust, scalable piece of software that runs every bit as well as rival commercial offerings from Microsoft and Netscape Communications. The competitive fallout from this deal is prompting the predictable yammering from industry analysts: What will happen to Microsoft and Netscape Web server products? What about IBM's line of Lotus Domino stuff? Yadda yadda. Yawn.

What makes this move noteworthy to corporate customers is the reassuring signal it sends that IBM's legendary resources and support will be just a phone call away.

Even more encouraging is IBM's public vow to join in that spirit by freely sharing its own improvements to Apache source code with the 'net community. That willingness to give something back — instead of walking away after pocketing the technology — sends another kind of reassuring signal. Corporate giants and freeware fanatics may have a lot more in common than they ever suspected.

Longtime 'netizens have deplored the "invasion" of their network by the crass commercialism of the Web. But imagine what the combined strengths of the business and the 'net community could accomplish. Imagine the impact such a detente could have on product quality and — even more compelling — choices in the marketplace.

Maryfran Johnson, *executive editor*
Internet: *maryfran_johnson@cw.com*

FLATTENER #4

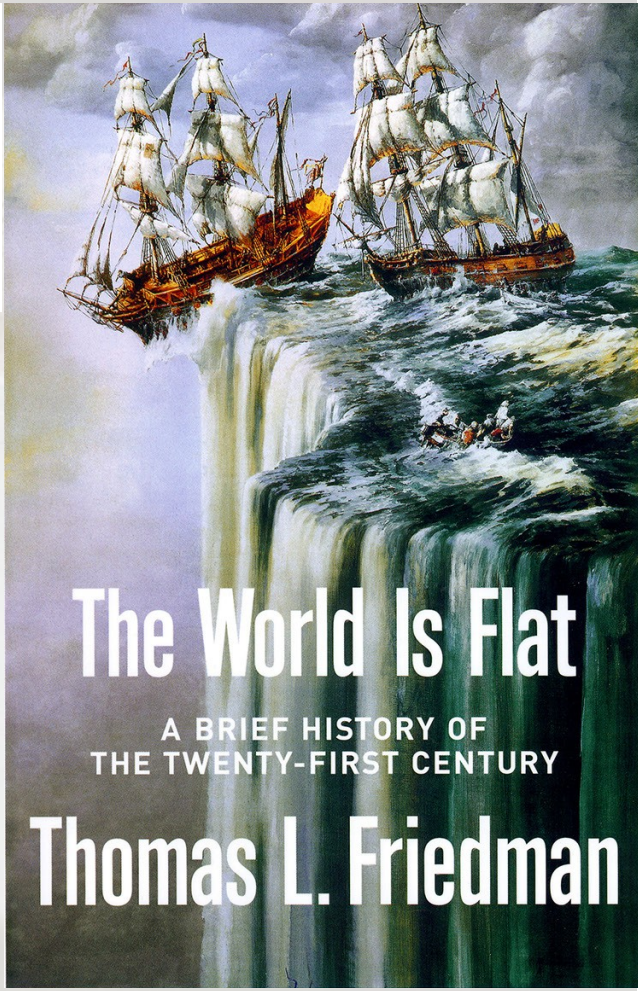
UPLOADING

Harnessing the Power of Communities

Alan Cohen still remembers the first time he heard the word "Apache" as an adult, and it wasn't while watching a cowboys-and-Indians movie. It was the 1990s, the dot-com market was booming, and he was a senior manager for IBM, helping to oversee its emerging e-commerce business. "I had a whole team with me and a budget of about $8 million," Cohen recalled. "We were competing head-to-head with Microsoft, Netscape, Oracle, Sun—all the big boys. And we were playing this very big-stakes game for e-commerce. IBM had a huge sales force selling all this e-commerce software. One day I asked the development director who worked for me, 'Say, Jeff, walk me through the development process for these e-commerce systems. What is the underlying Web server?' And he says to me, 'It's built on top of Apache.' The first thing I think of is John Wayne. 'What is Apache?' I ask. And he says it is a shareware program for Web server technology. He said it was produced for free by a bunch of geeks just working online in some kind of open-source chat room. I was floored. I said, 'How do you buy it?' And he says, 'You download it off a Web site for free.' And I said, 'Well, who supports it if something goes wrong?' And he says, 'I don't know—it just works!' And that was my first exposure to Apache . . .

"Now you have to remember, back then Microsoft, IBM, Oracle, Netscape were all trying to build commercial Web servers. These were huge companies. And suddenly my development guy is telling me that he's getting ours off the Internet for free! It's like you had all these big corporate executives plotting strategies, and then suddenly the guys in the mail room are in charge. I kept asking, 'Who runs Apache? I mean, who are these guys?'"

Yes, the geeks in the mail room are deciding what software they will be using—and what you will be using too, because communities of geeks are now collaborating to design new software and then to upload it to the world. It's called community developed software. But, thanks to the flat-



IBM invested $1B in *Linux* in 2001, claiming it recouped by 2002

COMPUTERWORLD October 29, 2001

COLIN JOHNSTON

Big Blue is making a big commitment to Linux across its product line. But will users bite?

By Mark Hall

IF IBM GETS ITS WAY, users will soon be thinking about operating systems the way investors view pork bellies: as mere commodities. The instrument the company will use to make this sea change in IT? Linux.

The implications of IBM's strategy for corporate IT planners are enormous. It affects everything from in-house development projects to server deployments. And the impact on IBM's competitors could be even more dramatic, say analysts and users.

Enterprise IT managers contemplating a move to Linux say they have much at stake. "We're putting 700 users on a mail system on top of Linux," says Dave Ennen, technical support manager at Winnebago Industries Inc. in Forest City, Iowa. "It's mission-critical."

Ennen's company is in the midst of a major server consolidation effort, taking advantage of Dallas-based Bynari Inc.'s InsightServer groupware application for Linux on an IBM zSeries mainframe. Ennen says the move eliminates the need for 40 Intel-based Windows servers that would have had to be upgraded to Microsoft Exchange and would have required a half-dozen support staff "instead of one or two for the mainframe."

Winnebago, a maker of motor homes and RVs, will run nine or 10 instances of Linux with InsightServer under IBM's virtual machine environment, which permits multiple operating systems to be managed on one system.

That's a big shift within IT away from having scads of Windows servers, but it's also a big hit on the competition. Not only would those 40 servers have run Exchange, but they also would have used Microsoft Windows 2000 Server and SQL Server software. Instead, the corporate e-mail application is running on an all-Big-Blue system.

"Microsoft has caused a lot of grief to IBM over the years. IBM sees Linux as a way to free itself from paying homage to Microsoft," says Al Gillen, an analyst at Framingham, Mass.-based IDC.

To be successful in the long run, however, IBM must assuage the skeptics regarding its new role as a contributor in the open-source community that wants to nurture Linux. And, more important, it must give Linux the enterprise-class capabilities IT managers expect from a data-center operating system.

TOP-DOWN APPROACH

As recently as 1998, IBM had no plans to become a provider of Linux-based products. In August of that year, it shipped the WebSphere Application Server, which runs on the open-source Apache Web server and Linux. It also launched an internal study on Linux's role in the IT industry.

IBM's first Linux technology came in December 1998 from an unsanctioned effort by IBM programmers in Germany, who ported the Linux kernel to the System/390 (now the zSeries mainframe) in their spare time, according to Dan Frye, director of the IBM Linux Technology Center.

Since then, the company has released Linux as an optional operating system on all of its servers. It now runs everything from IBM's DB2 and Domino Notes server software to encryption coprocessors and Tivoli Systems Inc.'s management utilities.

IBM's vice president of Linux, Steve Solazzo, says that what makes Winnebago and other companies choose Linux is that "it turns the operating system into a commodity that can run on any hardware."

This argument has increasing appeal to both independent software vendors like Bynari and users such as Gerry Sztabnik, director of middleware operations at the Security Industry Automation Corp. (SIAC) in New York.

Sztabnik recently completed porting SIAC's brokerage notification application from a Sun Solaris environment to Linux on IBM hardware in two and a half days. Because Linux is Unix at its core, Sztabnik says, porting code from Solaris to Linux is a snap. The underlying C code doesn't need to be modified, and most of the work entails recompiling on new hardware, in his case an IBM zSeries mainframe.

Had Winnebago opted for Exchange on Windows or had SIAC stuck with Solaris, they would have had only one hardware choice on which to run their applications in the future. But with IBM's full product line supporting Linux, IT managers can move their programs from low-end Intel servers to midrange systems or mainframes.

What's more, Sztabnik says, broader industry acceptance of Linux means that SIAC isn't even dependent on IBM as its sole system supplier. For example, both Hewlett-Packard Co. and Compaq Computer Corp. back Linux on their RISC and Intel servers.

As Winnebago's Ennen observes, "If we decided to move the Bynari app off the mainframe, we can do it."

In fact, that's what Paul Watkins did — in reverse. The network analyst at Newell Rubbermaid Inc. in Freeport, Ill., initially used his open-source multirouter traffic grapher performance-management tool on a low-end Intel server. But when it didn't give him the response time he needed, he moved it over to the company's System/390 running Linux.

"Now it just flies," Watkins says.

COMMUNITY ISSUES

As the crown prince of proprietary operating systems, IBM still faces skeptics among open-source developers, says Michael Tiemann, chief technology officer at Red Hat Inc., which supports many of IBM's Linux efforts.

And the competition is even less trusting. "Yes, they do want to commoditize the operating system," says Andy Ingram, vice president of Solaris at Sun Microsystems Inc. "They want to reduce the operating system to the lowest common denominator because that will drive more integration work for their Global Services division."

At Microsoft Corp., doubts go beyond IBM to Linux itself. Doug Miller, director of competitive strategy for the software giant, says he thinks Linux isn't a long-term bet for the data center. "I just don't see it taking over the world," he says.

Miller argues that IT doesn't buy servers and operating systems but rather business applications, which are predominately on Windows.

IBM is well aware of that skepticism and the current lack of data-center-specific software and services, acknowledges distinguished engineer Sheila Harnet, who works on Linux full-time at IBM.

Harnet says she thinks IBM has carefully picked domains where it could credibly contribute to making Linux a stronger enterprise operating system, such as in scalability, print services, file systems, volume management, serviceability and other high-end areas.

And that approach makes sense to IT professionals because, as Ennen puts it, "IBM puts some beef behind Linux." ◗

What's At Stake?

IBM is making significant investments in Linux – **more than $1 billion this year alone**, says Steve Solazzo, the company's vice president of Linux – and that amount will grow in 2002. IBM has **more than 250 developers working full time** on Linux and open-source projects and has **thousands of employees,** from sales to Global Services, trained on the operating system. Solazzo says Linux "is a **game-changer**. It will change the balance of power in the industry."

THE BUZZ

How open-source developers view IBM's role in Linux and other open-source projects: "I would say that the open-source community is fundamentally skeptical. But one of the properties of a good skeptic is that you can convince them."

– *Michael Tiemann, chief technology officer, Red Hat*

How users see IBM's Linux contributions: "IBM helps convince companies and their management that Linux is not just a hacker's tool."

– *Paul Watkins, network analyst, Newell Rubbermaid*

What the competition thinks about IBM's operating-system commodity strategy: "OSs are in one sense already a commodity. They all cost zero, being bundled in with the hardware they buy. In another sense of commodity, like bacon or oranges, where all producers make essentially the same thing, a Linux strategy can't possibly be motivated by commoditization, since Linux and [Windows] aren't interchangeable in the way that oranges from two growers are."

– *James Gosling, vice president and Sun fellow, Sun Microsystems*

Quick Link

- Is IBM's commitment to Linux good for users? Join the debate online: **www.computerworld.com/q?a1180**
- See what the Free Software Foundation has to say: **www.computerworld.com/q?24147**
- For more on Linux, visit our Operating Systems Knowledge Center: **www.computerworld.com/q?k1500**

IBM Roils Linux Waters

EVOLUTION OF IBM'S OPEN-SOURCE STRATEGY

1998

August 1998: IBM announces that its WebSphere e-business software will be based on the open-source Apache Web server application.

October 1998: Scientists at IBM's Academy of Technology recommend that top management review Linux's impact on the industry. Executives approve a formal study.

December 1998: A skunk-works project inside IBM's labs in Germany results in Linux code running on the System/390.

1999

May 1999: IBM announces DB2 database support for Linux.

August 1999: Launches open-source development team of five programmers.

November 1999: Announces port of Lotus Domino to Linux.

January 2000: Begins work on Linux journal file system, its first major official open-source project.

2000

March 2000: Team releases its first contributions, Linux drivers for Token Ring.

July 2000: IBM announces $200 million Linux development program for Europe.

July 2000: Makes Linux-based BlueDrekar middleware code available for the Bluetooth wireless specification.

August 2000: Launches $200 million Linux development program for Asia-Pacific. Signs distribution deal with Red Hat.

2001

January 2001: Announces that it will invest $300 million in Linux-based e-business services.

March 2001: Announces a broad range of storage products and services using Linux, including Shark-class storage systems on the z900, Modular Storage Server for midrange systems and Ultrium tape backup systems.

May 2001: Releases Linux for its iSeries midrange servers.

August 2001: Ships high-performance cryptographic coprocessor for Linux; unveils broader support for Linux in Tivoli security and Web management software.

IBM Systems Journal (2001): Communities of practice, Knowledge Management

Communities of practice and organizational performance

by E. L. Lesser
J. Storck

As organizations grow in size, geographical scope, and complexity, it is increasingly apparent that sponsorship and support of communities of practice—groups whose members regularly engage in sharing and learning, based on common interests—can improve organizational performance. Although many authors assert that communities of practice create organizational value, there has been relatively little systematic study of the linkage between community outcomes and the underlying social mechanisms that are at work. To build an understanding of how communities of practice create organizational value, we suggest thinking of a community as an engine for the development of social capital. We argue that the social capital resident in communities of practice leads to behavioral changes, which in turn positively influence business performance. We identify four specific performance outcomes associated with the communities of practice we studied and link these outcomes to the basic dimensions of social capital. These dimensions include connections among practitioners who may or may not be co-located, relationships that build a sense of trust and mutual obligation, and a common language and context that can be shared by community members. Our conclusions are based on a study of seven organizations where communities of practice are acknowledged to be creating value.

Steve walked into the meeting room and quickly grabbed a seat. Having just recently joined the company, his boss had recommended that he attend this weekly lunchtime meeting of VisualBasic programmers. He felt it was a good way for Steve to get "plugged into" the company, and would give him an opportunity to see some of the projects that others, across the firm, were working on.

The meeting began with a series of short introductions around the table. Then, one of the senior designers, Cindy, plugged a laptop into the overhead projector and started demonstrating a new set of programming tools that had been developed by one of the company's strategic partners. Steve took notice of the extended functionality of some of the tools, and saw an opportunity to use it on one of the new projects he would be spending time on. At the end of the meeting, Steve walked up to Cindy and introduced himself. They spent another 20 minutes discussing the opportunity to use the tools on Steve's project and how Steve might be able to also use some existing code from one of Cindy's recent development efforts. After writing down Cindy's contact information on a napkin, he headed back to his office, thinking about this new course of action.

As organizations grow in size, geographical scope, and complexity, it is increasingly apparent that sponsorship and support of groups such as the one described above is a strategy to improve organizational performance. This kind of group has become known as a community of practice (CoP)—a group whose members regularly engage in sharing and learning, based on their common interests. One might think of a community of practice as a group of people playing in a field defined by the domain of skills and techniques over which the members of the group interact. Being on the field provides members with a sense of identity—both in the individual sense and in a contextual sense, that is, how the individual relates to the community as a whole.[1] A

The knowledge management puzzle: Human and social factors in knowledge management

by J. C. Thomas
W. A. Kellogg
T. Erickson

Knowledge management is often seen as a problem of capturing, organizing, and retrieving information, evoking notions of data mining, text clustering, databases, and documents. We believe that this view is too simple. Knowledge is inextricably bound up with human cognition, and the management of knowledge occurs within an intricately structured social context. We argue that it is essential for those designing knowledge management systems to consider the human and social factors at play in the production and use of knowledge. We review work—ranging from basic research to applied techniques—that emphasizes cognitive and social factors in knowledge management. We then describe two approaches to designing socially informed knowledge management systems, social computing and knowledge socialization.

Knowledge management (KM)—also known under rubrics such as organizational learning, organizational memory, and expertise management—has received increasing attention over the last decade. Indeed, it is fair to say that knowledge management is well on the way to becoming a distinct field, with its own theories, jargon, practices, tools, skills, and other accoutrements of an independent discipline. This paper is motivated by our concern that the codification of knowledge management is proceeding a little too rapidly, and that we may end up with a conception of knowledge management that is too neat and too simple to survive in the wilds of the workplace.

The dominant conception of knowledge management—particularly that which has spread beyond the circle of researchers and practitioners into the marketplace—is overly tidy. Knowledge management is seen primarily as a problem of capturing, organizing, and retrieving information, evoking notions of databases, documents, query languages, and data mining. Knowledge is seen as passive, analytic, and atomistic: it is composed of facts that can be stored, retrieved, and disseminated, with little concern for the context in which the facts were originally embedded, and little concern for the new and often quite different contexts in which they will be used. In this view, as one widespread advertisement recently claimed, knowledge management is nothing more than getting the right information to the right people at the right time.

This is a nice picture, but one with which we are not comfortable. Whereas there is no denying the importance of factual knowledge and the usefulness of information technologies, we believe that there are many other issues that are of critical import. Our goal, therefore, is to bring forward a set of results—ranging from basic research findings to practical techniques—that we believe to be very relevant to knowledge management, even as they are at risk of being left out of the KM picture. Overall, our strategy in this paper is to back away from a coherent picture of knowledge management. We suggest that it is more valuable to see knowledge management as a puzzle, especially if we focus on the puzzle pieces: our basic approach will be to add a number of new

The IBM GIO 2.0 (2006) explored the nature of innovation

The Global Innovation Outlook provides a platform for some of the world's most interesting thinkers—provocateurs and pragmatists alike—to engage in a series of open, candid and freewheeling conversations about important issues of our day, from healthcare to the environment, the role of government to the future of the enterprise. Rather than predicting the future, it is a search for the sparks that will ignite meaningful change for individuals, businesses and the world.

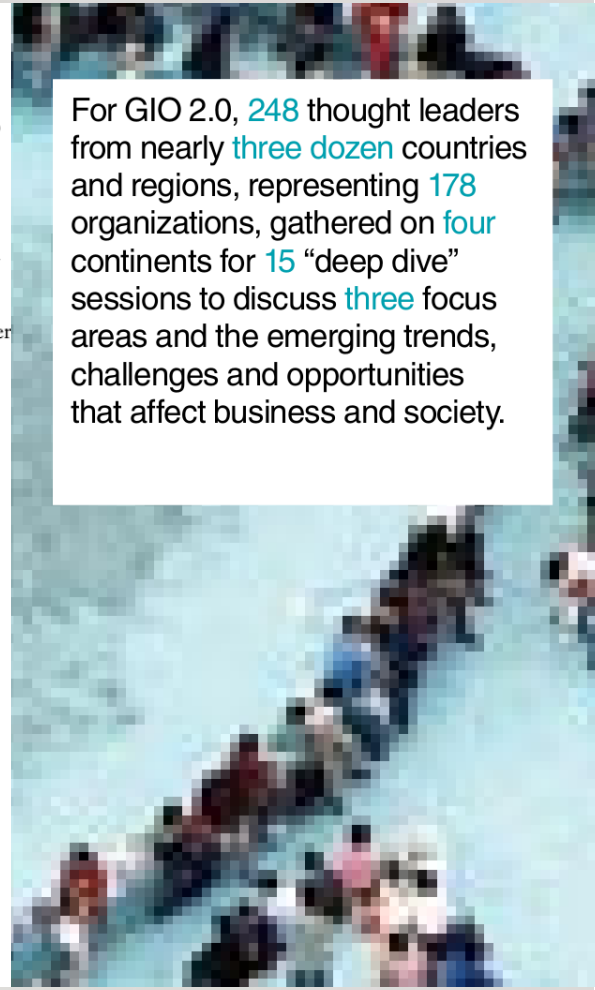
The GIO also investigates innovation itself—and the profound ways in which it is changing. In fact, the most essential finding of the first GIO, which was conducted in 2004, might be that innovation is no longer invention in search of purpose, no longer the domain of a solitary genius looking to take the world by storm. Instead, innovation is increasingly:

Global. The widespread adoption of networked technologies and open standards is removing barriers of geography and accessibility. Anyone and everyone can participate in the innovation economy.

Multidisciplinary. Because the challenges before us are more complex, innovation now requires a diverse mix of talent and expertise.

Collaborative and open. More and more, innovation results from people working together in new and integrated ways. Within this collaborative environment, notions of intellectual property are being re-examined. And those entities that view intellectual assets as "capital" to be invested and leveraged—rather than "property" to be owned and protected—will likely reap the greatest returns.

Perhaps that's why the GIO proves to be such a compelling exercise for IBM and our ecosystem partners. It is an investigation that invariably uncovers themes or patterns that transcend particular industries or interests. Ultimately, the GIO is an investigation of innovation that matters for us all.



For GIO 2.0, 248 thought leaders from nearly three dozen countries and regions, representing 178 organizations, gathered on four continents for 15 "deep dive" sessions to discuss three focus areas and the emerging trends, challenges and opportunities that affect business and society.

In the *21st century*, the *nature of innovation* is increasingly...

Open

Collaborative

Multidisciplinary

Global

"Innovation as open, collaborative, multidisciplinary, global" | June 13, 2008 at http://coevolving.com/blogs/index.php/archive/innovation-as-open-collaborative-multidisciplinary-global/

An inferred shift from *Industrial Age innovation* educates

Industrial Age		*21st Century*
Private methods and development enabling autonomous control over designs	strategy	**Open** standards and interfaces leveraging expedient platforms for advancing design
Transactional production chains linked by inter-organizational contracting	relationship	**Collaborative** alliances coproducing accelerated learning
Analytical problem-solving	method	**Multidisciplinary** conversations
Colonial trade	economics	**Global** talent

"Innovation as open, collaborative, multidisciplinary, global" | June 13, 2008 at http://coevolving.com/blogs/index.php/archive/innovation-as-open-collaborative-multidisciplinary-global/

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

a. Open Innovation Learning
b. Open Innovation Learning
c. *Descriptive* theory building
d. *Normative* theory building
e. *OIL* + Systems Thinking

Three *descriptive theory building* streams are alongside 3 *paradigms*

Paradigm:
Architectural problem seeking

Theory building:
Quality-generating sequencing

Paradigm:
Inhabiting disclosive spaces

Theory building:
Affordances wayfaring

Paradigm:
Governing subworlds

Theory building:
Anticipatory appreciating

With *architectural problem solving*, a theory of *quality-generating sequencing*

Theory building:

Quality-generating sequencing

- Generative codes (Christopher Alexander)
- Structural quality (elaboration of form as horizontal) vs. dynamical quality (elaboration of organization as vertical)
- Unfolding wholeness over time
- Cross-scale interactions (pacing layers)
- Patterns concerns entailed:
  - Program envisioning
  - Program realizing
  - Program elaborating

Paradigm:

Architectural problem seeking

- Morphogenesis
- Articulating space (dividing into parts, putting together by joints)
- Autopoietic (self-reproducing) or allopoietic (produced by something external to the self)
- Problem-seeking (wicked problems c.f. problem-solving)

With *inhabiting disclosive spaces*, a theory of *affordances wayfaring*

Theory building:

Affordances Wayfaring

- Affordances as complementarity of an animal and its environment, furnishing an invariant meaning
- Wayfaring as embodied experience of living through, around, to and from places
- Attentional in a labyrinth c.f. intentional in a maze)
- Material entities, recognized as boundary objects
- Patterns concerns entailed:
  - Enskilling
  - Equipping
  - Legitimating

Paradigm:

Inhabiting disclosive spaces

- An organized set of practices for dealing with oneself and the world
- Dwelling, dissolving distinctions between occupying and building
- Worlds not shared, as customary skills not appropriate everywhere
- Taskscapes in the temporality of work practices (c.f. dwelling on the land on landscapes)

With *governing subworlds*, a theory of *anticipatory appreciating*

Theory building:

Anticipatory appreciating

- Appreciating model as norm-seeking
  (c.f. rational model of goal-seeking)
- Anticipatory behaviour as changes in a system in the present, caused by events that have not yet happened, but entailed in the future
- Patterns concerns entailed:
  - Judging material reality
  - Judging formal value(s)
  - Judging efficient instrumentality

Paradigm:

Governing subworlds

- Moral syndromes,
  commercial and guardian
- Order regulated by forces
  (within, as self organization;
  without as environmental constraints)
- Subworlds as local elaborations of a commonsense world we share
- Governing as setting and enforcing bounds
  (c.f. managing the conduct of an enterprise or organization)

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

a. Open Innovation Learning
b. Open Innovation Learning
c. *Descriptive* theory building
d. *Normative* theory building
e. *OIL* + Systems Thinking

Normative theory defines "what causes the outcome of interest"

J PROD INNOV MANAG 2006;23:39–55

THE JOURNAL OF PRODUCT INNOVATION MANAGEMENT

The Ongoing Process of Building a Theory of Disruption

Clayton M. Christensen

The easiest way to respond to the critiques and complements the other authors in this issue have written about the model of disruption would simply be to address them head on—to accept some as useful additions or corrections and to suggest that others are ill-founded. Because this special issue of *JPIM* represents a unique opportunity to examine the process of theory-building as it unfolds, however, this article is structured in a way that addresses the other scholars' suggestions in the context of a model of the process by which theory is built and improved. My hope in doing so is that this issue might not just be an examination of this particular theory of disruptive innovation but that it might also constitute a case study about theory-building itself—a study that can help scholars of management in different fields to conceptualize how the theory-building process is or is not at work in their domain—and how they might help the process work better.

A Model of the Theory-Building Process

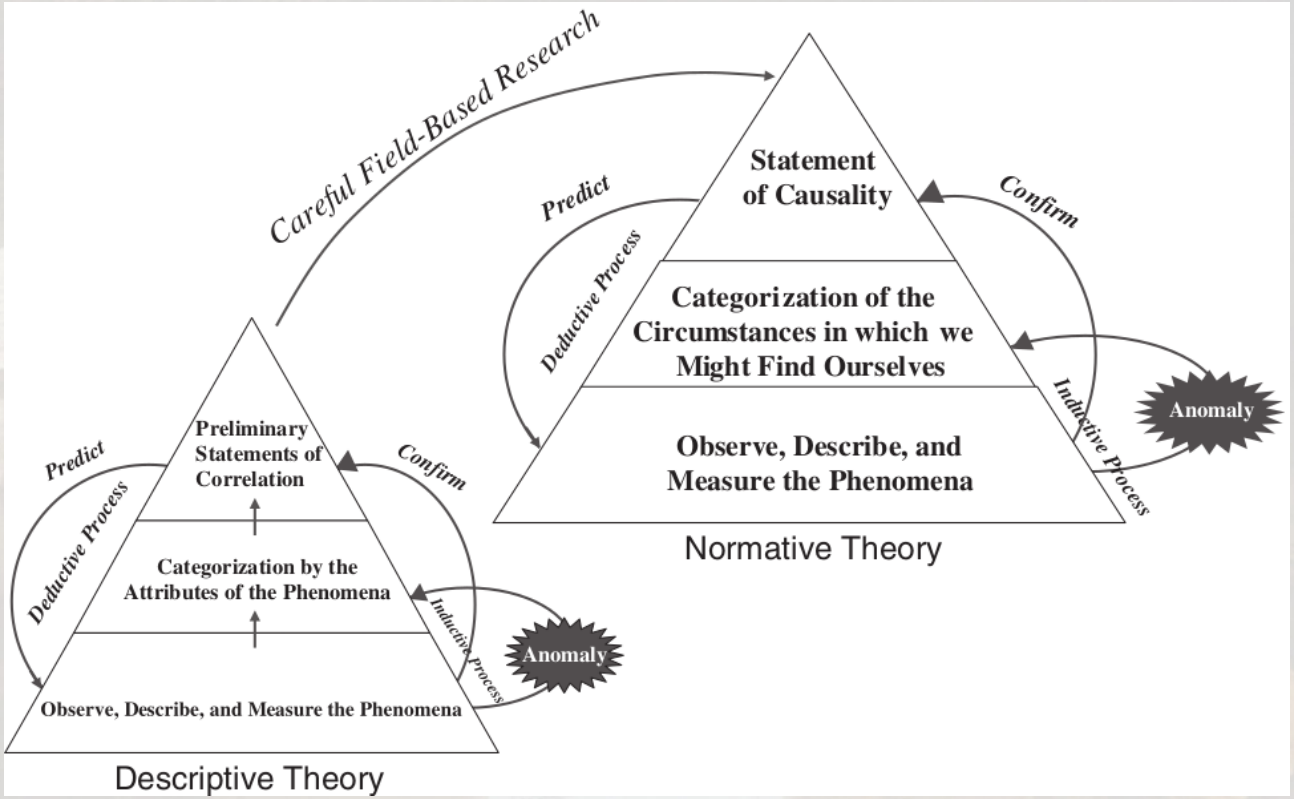
Some years ago in a doctoral seminar my students and I examined how communities of researchers in a variety of disciplines had cumulatively built bodies of understanding. Seeing some stunning commonalities in the processes these scholars had followed, we synthesized a model of the process of theory building (for a summary, see Carlile and Christensen, 2005). My students and I found this model extremely useful as we designed our own research, positioned our work within streams of prior researchers' efforts, and evaluated the reliability and validity of various papers. The present article recounts the development of the theory of disruption within the context of this model of what theory is and how it is built. It also suggests how the comments of the other authors in the current issue of *JPIM* might contribute to the improvement of this body of theory. In this way, I hope that both the content of this theory and the process by which it is being built might become clearer.

Our model asserts that theory is built in two major stages: the descriptive stage and the normative stage. Within each of these stages, theory builders proceed through three steps. The theory-building process iterates through these three steps again and again. In the past, management researchers have quite carelessly applied the term *theory* to research activities pertaining to only one of these steps. Terms such as *utility theory* in economics and *contingency theory* in organization design, for example, actually refer only to an individual step in the theory-building process in their respective fields. It is more useful to think of the term *theory* as a body of understanding researchers build cumulatively as they iterate through each of the three steps in the descriptive and normative stages. This should be abundantly clear as we examine the theory of disruption. It already has evolved considerably as a growing group of scholars, including those whose work is published herein, have worked to refine it. Among the most notable improvements to date have been Adner and Zemsky (2003), Adner (2002), Gilbert (2001), Christensen and Raynor (2003), and Christensen, Anthony, and Roth (2004).

Building Descriptive Theory

The descriptive stage of theory building is a preliminary stage because researchers generally must pass through it before developing normative theory. The three steps researchers use to build descriptive theory are observation, categorization, and association.

Address correspondence to: Clayton M. Christensen, Harvard Business School, Boston, MA 02163. E-mail: cchristensen@hbs.edu.



- Descriptive theory produces "statements of correlation".
- An "understanding of causality enables researchers to assert what actions managers ought to take to get the results they need".

Normative theory on *Innovation Learning* may guide emerging cases

Innovation Learning with the rise of:

Polycentric Governance

- Deglobalization, Brexit, Trump presidency
- International innovation as:
  - i) complete concentration; or
  - ii) core-periphery concentration; or
  - iii) sequential dispersal; or
  - iv) modularized dispersal; or
  - v) inclusive dispersal.

Innovation Learning with the rise of:

The Internet of Things (IoT)

- Physical world interweaved with actuators, sensors + computational elements through network connectivity
- Smart cities
- Smart homes
- Smart grid
- Smart buildings
- Smart transportation
- Smart health
- Smart industry

Innovation Learning with the rise of:

Cognitive Computing

(Intelligence Augmentation)

- An evolution from
  - mechanical tabulating era (1900s-1940s); to
  - digital programming era (1950s to present); to
  - cognitive era (2011, IBM Watson winning Jeopardy).
- Man-machine symbiosis in cooperative interaction
- Open AI
- Partnership on AI

Three *normative theory building* streams are alongside one *paradigm*

Paradigm:

Co-responsive movement

- Ecological anthropology: getting a grip on the larger world
- Material culture studies: artifacts with physicality + history with associated human beings

Theory building:

Innovation learning for

- Enskilling attentionality
- Episteme (know why)

Theory building:

Innovation learning by

- Weaving flows in form-giving
- Techne (know how)

Theory building:

Innovation learning alongside

- Agencing strands
- Phronesis (know whom, when, where)

Innovation learning for: *enskilling attentionality* as 3 types

Paradigm:

Co-responsive movement

Theory building:

Innovation learning **for**

- Enskilling attentionality
- Episteme (know why)

Type: **Proto-learning**
- Selecting an alternative in context

Type: **Deutero-learning**
- Changing the set or sequence of alternatives in contextual change

Type: **Trito-learning**
- Changing systems of alternatives in meta-contextual change

Innovation learning by: *weaving flows in form-giving* as 3 types

Paradigm:

Co-responsive movement

Theory building:

Innovation learning **by**

- Weaving flows in form-giving
- Techne
  (know how)

Type: **Learning-by-doing**

- Accumulating experience, in both organizational + personal senses

Type: **Learning-by-making**

- Constructing with sociomaterial creativity, in critical making

Type: **Learning-by-trying**

- Co-configuring architecturally + dialogically, social interaction + technology

Innovation learning alongside: agencing strands as 3 types

Paradigm:

Co-responsive movement

Theory building:

Innovation learning alongside

- Agencing strands
- Phronesis
  (know whom, when, where)

Type: **Polyrhythmia entangling eurhythmia**

- Experience in living beings

Type: **Regenerating entangling preserving**

- Continuity in living nature vs. form

Type: **Less-leading-to-more entangling more-leading-to-more**

- Increasing complicatedness or complexity

Teleonomy learns from teleology in a philosophy with alternative stable states

Teleology:
Goals, objectives, ideals

- Emphasis on *final cause*, of Aristotle's four causes:

(i) material cause (that out of which);
(ii) formal cause (the account of what it-is-to-be);
(iii) efficient cause (the primary source of change or rest);
(iv) **final cause** (the end, that for the sake of which a thing is done).

Teleonomy:
Environmental change, somatic (cellular) change, genotypic change

- A process or behaviour which owes its goal-directedness to the operation of a program
- Coded or prearranged information that controls a process (or behaviour) leading it toward a given end.

Alternative stable states:
Panarchy, resilience, regime shifts

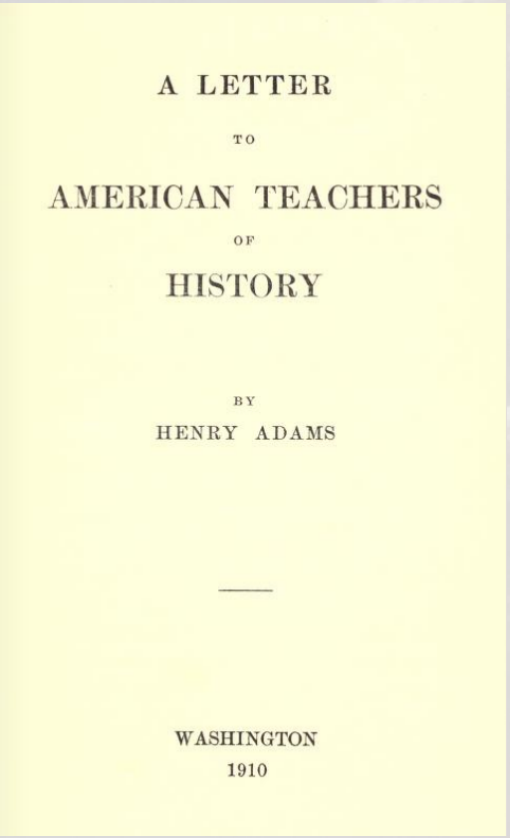
- From community ecology, changes in state variables (e.g. population densities).
- From ecosystem ecology, changes to the parameters governing interactions within an ecosystem.

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception

a. Open Innovation Learning
b. Open Innovation Learning
c. *Descriptive* theory building
d. *Normative* theory building
e. *OIL* + Systems Thinking

“Stable equilibrium is death”. Is *innovation learning* a living system?


A LETTER

TO

AMERICAN TEACHERS

OF

HISTORY

BY

HENRY ADAMS

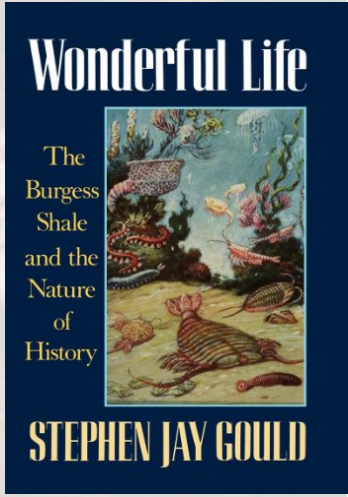
WASHINGTON

1910

… if one physical law exists more absolute than another, it is the law that **stable equilibrium is death**.

A society in stable equilibrium is — by definition, — one that has history, and wants not historians. [Adams, p. 186]

… Gould has shown that evolution has been by catastrophes, like the one that caused the demise of the dinosaurs and more serious ones that extinguished up to percent of all species nearly six hundred million.
Gould has concluded that such catastrophes have been more instrumental in shaping the course of evolution than competition and natural selection.
If so, then no necessary direction can be imputed to evolution, and **the current state of nature may not be inevitable and predictable**. [Burich p. 645]



Adams, Henry. 1910. A Letter to American Teachers of History. Washington [Press of J.H. Furst]. http://archive.org/details/alettertoamerica00adamuoft.

Burich, Keith R. 1992. “‘Stable Equilibrium Is Death’: Henry Adams, Sir Charles Lyell, and the Paradox of Progress.” The New England Quarterly 65 (4): 631–47. doi:10.2307/365825.

“Stable equilibrium is death” at https://stream.syscoi.com/2017/09/24/stable-equilibrium-is-death/

Is your (theory building) system generative?

Systematic	*Systemic*
Somatic (adaptive, cellular) change	Genotypic (generational) change
Non-living, effect-producing (allopoietic)	Living, systems-generating (autopoietic)
Reactive	Co-responsive

Agenda

1. Introductions
2. Outline
3. Commentary
4. Q&A
5. Reception