Notes from Steven Weber's "The Success of Open Source"

Background

Sakai Board Chair Joseph Hardin recommended Weber's book because of its clear analysis and relevance to the Sakai Project and the emerging Sakai Educational Partners Program.

Weber, Professor of Political Science at the University of California, Berkeley, uses the perspective of the social organization of cooperation and production that characterizes open source software. He describes "open source software" "as an experiment in social organization around a distinctive notion of property."[1] He observes "Open source software is an odd mix of overblown hype and profound innovation."[2]

Weber's Model

Weber cites Eric Raymond's explanation of open source development based on the "gift economy."[3] Raymond says that "gift economies are adaptations not to scarcity but to abundance." Weber writes: "When you are this far out on a curve of diminishing marginal returns to owning something, what makes sense is to give things away (and gain status) instead."

Earlier Goldhaber advanced "Attention Economy" which could also explain why some developers participate in open source.[4] They gain attention that can benefit them; for example, higher consulting fees or higher faculty position and salary.

Weber acknowledges Metcalfe's Law—"positive network externalities, meaning that the value of the network increases disproportionately as it grows."[5] And Moore's law—"Rapid increases in processing power at declining prices." These two combine to explain why corporate-sponsored developers would cooperate on the development of open standards that later leads to product competition. And why open source software developers want their software to be widely implemented.

Open Source Software and Competition

Citing the value of compilers, Weber writes: "Precompetitive collaboration: Competitors share early stages of research that benefit all."[6] This is the theme of Steve Holbrook,

---

[1] Chapter 1, page 1. Steven Weber, "The Success of Open Source," Harvard University Press, Cambridge, Massachusetts, 2004, ISBN 0-674-01292-5.
[2] Page 7.
[3] Eric S. Raymond, "Homesteading the Noosphere," *First Monday* vol. 3, no. 10 , October 5, 1998, p. 99.
[4] Michael Goldhaber, "The Attention Economy and the Net," First Monday vol 2 No. 4, April 7, 1997.
[5] Page 248.
[6] Page 21.

Program Director of Emerging e-Business Standards for IBM Corporation in his keynote at the May 2003 PESC Annual Conference.[7] He he explained why IBM would cooperate closely with Microsoft in the development of the standards underlying Web Services in order to create a large market that would benefit both companies. IBM Vice President of Intellectual Property and Standards Jim Stallings had a similar description of IBM's support of open source at the December 2004 "Open Source Summit." [8]

The Legal Context

"Open source developers are some of the most vehement defenders of intellectual property rights. … Open source collaboration depends on an explicit intellectual property regime, codified in a series of licenses." "The principle goals of the open source intellectual property regime is to maximize the ongoing use, growth, development, and distribution of free software. To achieve that goal, this regime shifts the fundamental optic of intellectual property rights away from protecting the prerogatives of an author toward protecting the prerogatives of generations of users."[9] Here Weber advances the same argument—public good—that created "open content" and led the U.S. government to place its documents and much of its software in the public domain.

Weber failed to note that many of the successful "open source" projects—MySQL, Sendmail, and LAMS are examples—retain and license commercial rights. And the success of O'Reilly Media Inc. depends upon open source's inadequate documentation and copyright.

Organization of Open Source Development

"Conway's Law argues that the structure of a system—in this case, a technological system of software code—mimics the structure of the organization that designed it. Because the point of organization ultimately is to facilitate successful coordination of the technology development process, Conway's Law has been interpreted to mean that the technology architectures should drive thinking about the organization, not vice versa."[10]

This becomes a significant factor if the modularization of the software design—necessary for open source development of complex systems—does not match the organization. This match becomes even more difficult when the organization is being transformed as education in adapting to information technology.

---

[7] See his slides from the 5th Annual Conference of the Postsecondary Electronic Standards Council held May 6-8, 2003 in Alexandria, Virginia. The are available at www.immagic.com/ELIBRARY/GENERAL/PESC/PESC0305.pdf.
[8] Hosted by The Sedona Conference and Conversations and the r*smart group, December 1-3, 2004, Scottsdale, Arizona.
[9] Page 84.
[10] Page 86.

Criteria for Success

Weber believes the success of an open source project should be market share. ”Pragmatism rules and the metric for success is now clearly laid down—it is about conquering the “market” by building software and software packages that meet the needs of users better than any alternatives. In short, a new production process with a very standard measure of achievement was established.”[11] Because of the network effect, this means the open source software “product” must dominate its market. The market may be a specialized segment that has unique needs rather than a large generic market. Referring to IBM’s support of Red Hat’s Linux operating system: “If the open source movement measured its success partly by whether it could achieve status as a competitive technological solution for mainstream business use, by 2000 that question was answered definitely in the affirmative.”[12] What he did not say was that few open source projects are successful by this measure.[13]

In August 1998 Weber quotes Microsoft’s Vinod Valloppillil “…open source as a development process that was ‘long term credible’ and a ‘direct, short-term revenue and platform threat to Microsoft.’ It confessed that ‘commercial quality can be achieved/exceed by OSS projects.’”[14]

Weber notes in 1998 Eric Von Hippel observed “…users, when they are empowered to do so by technology and by the legal and economic structures in which that technology is embedded, will innovate more quickly and effectively that will [software] manufacturers.” “Lead users face needs that will likely become general in the market, but they face them sooner than everyone else; and they are positioned to benefit in significant ways if they can obtain an answer to those needs.”

Eric Raymond has commented that the developers of operating systems were also the users of operating systems. Bill Olivier at the December 2004 meeting of the JISC Enterprise SIG made a similar point when he commented there are two developer communities for a software product—the developers who are building the software and the users who will be using the software product. When these are the same community—the Linux operating system example—then communication is not an issue. However, for learning delivery and learning management systems, the users—faculty and students—are not the same as the developers. Olivier said Sakai may succeed developing an open source course management system because it has “lead users” in the Sakai Educational Partners Program. If they strongly influence the design of the Sakai products, then the project will be successful, more successful than commercial products. Here Olivier

---

[11] Page 116.
[12] Page 121.
[13] On December 20, 2004 Source forge listed 92,465 projects and 976,457 registered users. None of the top projects is a recognized software product; based on 2002 estimates there would be about 600 “mature projects” and less than 100 successful products. See Sandeep Krishnamurthy, “Cave or Community? An Empirical Examination of 100 Mature Open Source Projects,” *First Monday*, vol. 7, nr. 6, 3 June 2002.
[14] Page 126 quoting from “Open Source Software: A (News?) Development Methodology” often referred to as the Halloween document.

extends Weber's observations into another criterion that must be met by successful open source applications.

"Scratching your own itch"—cited by many as the source of open source software developer motivation—"then becomes a story about the satisfaction that comes with solving your own problem *and* solving someone else's problem as well.[15] Weber could have used Goldhaber's Attention Economy to explain why recognition would motivate a developer. His thesis is better developed that Eric Raymond's.

Weber observes self-selection of open source developers may occur because "If you are mediocre, the last thing you want is for people to see your source code."[16]

Referring to technical design, Weber observes "Source code modularization obviously reduces the complexity of the system overall because it limits the reverberations that might spread out from a code change in a highly interdependent and tightly coupled system. Clearly it is a powerful way to facilitate working in parallel on many different parts of the software at once."[17] Fred Brooks made a similar argument when he wrote that software complexity sharply lowers development productivity as a project gets larger.[18] Here the key to design is to have a number of separate software developments with minimum interdependencies. Weber may have identified another key factor in the success of open source development—deconstructing the system into modules that have the least interrelationship. This observation also suggests that software products that are hastily developed without adequate design will have higher maintenance costs than quality software developed more carefully.

Weber had another caution for open source development: "The technical aesthetic that the open source process depends on it more focused on solving complex programming problems than it is on engineering for convenient user interfaces and simplicity. … Some programmers believe that understanding direct user experience requires extensive physical interaction—that is focus groups, not Internet mailing lists. The more general point here is that design for components like user interfaces still rests on the transmission between individuals of a great deal of tacit information."[19] The Joint Information Systems Committee (JISC) has established projects to improve the user interface for many of its software developments. This has been successful with portal development and has now been extended to library and repository search portlets. JISC may have implemented a solution for a weakness of open source development Weber identified.

---

[15] Page 139.
[16] Page 143.
[17] Page 173
[18] Fredrick P. Brooks, "The Mythical Man Month," Essays on Software Engineering, 20th Anniversary Edition," Addison-Wesley Professional, 2 August 1995, ISBN 0201835959.
[19] Page 238.

Motivation for Open Source

Referring to the market and open standards, Weber comments "…releasing source code is a credible commitment against the possibility of using a standard for technological lock-in and hold up."[20] College and university representatives have cited this reason for participating in JA-SIG and Sakai. But it may be open standards is better response to lock-in than open source.

How important is software? "But software is the rate-limiting factor of the information economy and, as hardware capabilities continue to advance, the mismatch becomes increasingly evident." This suggests there is no limit to the scope and need for both open source and commercial software in the foreseeable future.

[20] Page 239.