Factors Leading to Success or Abandonment of Open Source Commons: An Empirical Analysis of Sourceforge.net Projects

Charles M. Schweik[1], Robert English[2] , Sandra Haire[3]

[1]Dept Natural Resources Conservation, Center for Public Policy and Administration, National Center for Digital Government, University of Massachusetts, Amherst, USA, cschweik@pubpol.umass.edu

[2] National Center for Digital Government, University of Massachusetts, Amherst, USA, renglish@gmail.com

[3]Dept Natural Resources Conservation, University of Massachusetts, Amherst, USA, shaire@nrc.umass.edu

Abstract

Open source projects are a form of common property regime or "commons" where developers collectively act and collaborate to develop software. Over the last decade significant theoretical and empirical advances have been made understanding open source commons, led by scholars such as Yochai Benkler (The Wealth of Networks) and Eric von Hippel (Democratizing Innovation). However, open source as an "ecosystem" is changing, moving from the all-volunteer, user-developer setting to a more complex collaborative environment involving not only volunteers but also employees of firms, nonprofit organizations and government agencies. Moreover, while some open source projects succeed from a collaborative standpoint, it is likely that the majority of them do not.

The general research question we are pursuing in this paper and the broader research project it is related to is: What factors lead to collaborative success or abandonment in open source commons? The answer has implications not only for informing other open source development efforts, but also "open content" collaborations outside of software. The paper is divided into two major parts. The first part reports our findings from a comprehensive review of literature from a variety of disciplines looking for factors which theoretically or empirically are thought to influence the success or abandonment of FOSS commons. The second section undertakes an empirical analysis of FOSS projects, testing factors embodied in project information acquired from the open source hosting site Sourceforge.net.

1. Introduction

In 2005 we began a study funded by the National Science Foundation to study open source collaborations – what we call "open source commons," since with their licensing, they are a form of common property regime. The overall goal of the study is to identify "design principles" that lead these projects toward successful collaborations rather than abandoned efforts. Since that time, we've conducted an extensive review of theoretical and empirical literature, interviewed open source developers, and are currently completing quantitative analysis of thousands of open source projects on Sourceforge.net. In this paper we summarize of some of the work we have completed so far.

2. Factors Thought to Influence Open Source Collaborations

We have reviewed a sizable amount of theoretical and empirical literature in a variety of disciplines searching for factors thought to contribute to the success or abandonment of open source collaborations. We started with the traditional information systems development literature, but then moved to literature on distributed work and virtual teams, as well as literature on collective action and commons governance and management. Much of this latter work focuses on collaborations in natural resource commons or common property, but very recently scholars are studying "digital commons," such as open access publishing, and open content collaboration (Hess and Ostrom, 2007). In this section we provide an overview of the variables we have identified through this process. Following Ostrom (2005), we organize them into physical, community and institutional attributes (Figure 1).

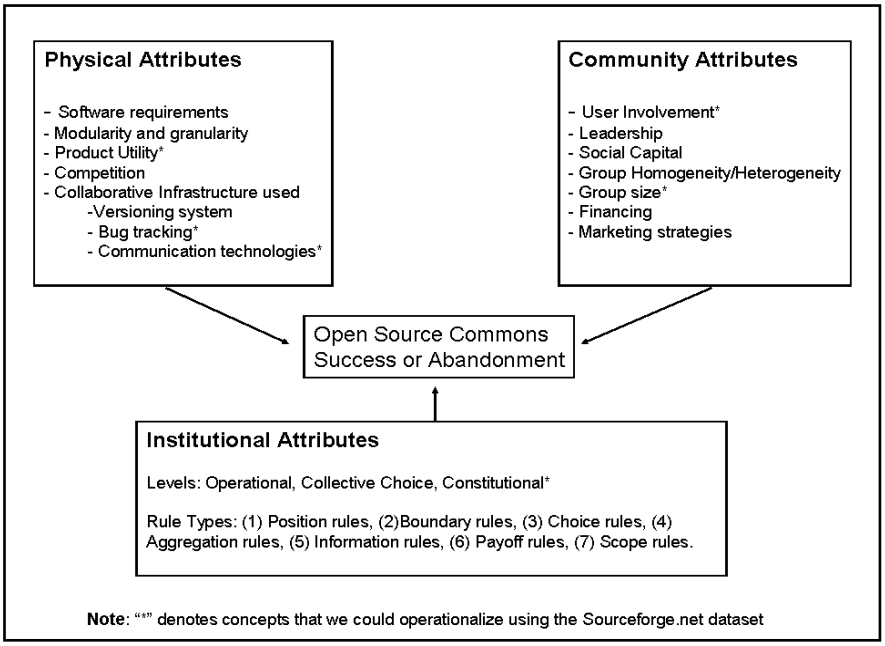

Figure 1. Factors Thought to Influence the Success or Abandonment of Open Source Collaborations

2.1 Physical Attributes of Open Source Commons

Physical attributes refer to the set of variables related to the actual software developed or some of the technological infrastructure needed for team coordination. Our review identified the following variables that potentially affect the success or abandonment of open source commons: (1) software requirements, (2) modularity, (3) product utility, (4) competition, and (5) collaborative infrastructure.

Software requirements refer to the processes used to determine what the software will or should do. It is thought that projects with clearly defined visions will do better than ones without such visions. *Modularity* has to do with the design of the software, and whether it is easily broken down into separate, relatively standalone components. Within limits, a modular design is thought to make it easier for contributors to "carve off chunks" of the project that they intend to work on (Weinstock and Hissam, 2005). *Product utility* describes the obvious; that a project will be more successful if the software being produced is something that people want or need (Ibid.). *Competition* refers to whether the project is unique in what it is trying to do, or whether there are other similar projects available. Significant competition would lead to potentially fewer available people or organizations wanting to join in to any particular project. Competition also captures the situation where a rival technology comes along that reduces people's interest in the product being developed. Finally, *collaborative infrastructure* describes the types of technologies used to help coordinate the collaborative team. There are a variety that may be used, including a code version control system, a bug tracking system, and a number of communication and documentation technologies (e.g., email lists, web-based forums, Internet Relay Chat, etc.). The particular configuration may be particularly important in reducing the time expended by team members. For instance, a norm emphasizing the use of a question and answer forum for help creates, at the same time, a searchable documentation database.

2.2 Community Attributes of Open Source Commons

This label captures variables related to the people who are developing the software, along with the financial and marketing aspects of the project. These include: (1) user involvement; (2) leadership; (3) social capital; (4) group homogeneity/heterogeneity; (5) group size; (6) project financing; and (7) marketing strategies.

User involvement is one of the long-standing variables known to influence the success or failure of traditional software development projects (Ewusi-Mensah, 2003) as well as in open source settings (von Hippel and von Krough, 2003; von Hippel, 2005). Similarly, the concept of *leadership* appears repeatedly in the literature and is thought to be a success or abandonment factor in both traditional face-to-face teams and virtual teams (Tyran et al., 2003). Components include how well the leader(s) are able to motivate the team, and how well goals are articulated (Katzenbach and Smith, 1993). In political science and economics, the degree of *social capital* – usually characterized as "trust" between community members – is often discussed when describing a "healthy" or vibrant community (Putnam 2007). In other commons settings three factors contribute to the establishment and maintenance of trust: reciprocal relationships (e.g., I help you, you help

me), repeated interactions (Ostrom, et al., 1999), and regular face-to-face meetings (Nardi and Wittaker, 2002).

Group heterogeneity is thought to influence the ability for a team to act collectively (Sandler, 2004). Varughese and Ostrom (1998) sub-divide the concept into three categories: (1) socio-cultural, (2) interest, and (3) asset heterogeneity. Socio-cultural heterogeneity includes attributes such as ethnicity, religion, gender, caste, language, or other cultural distinctions. The presumption is that groups with diverse socio-cultural backgrounds will have more difficulties working together because of a lack of understanding and, potentially, a lack of trust. Interest heterogeneity captures the motivations of people who participate in a commons. Volunteers often participate in open source for different reasons than some paid programmers. It is an open question as to whether diverse or diverging interests in open source affect collaboration, although some literature suggests that tensions can arise when volunteer and business interests coincide. Lastly, asset heterogeneity captures the idea that some individuals may bring to a project capabilities or resources that others on the team might not have themselves. For example, concepts like wealth and political power are two types of assets found in some commons settings. Studies related to natural resource commons have found that heterogeneity in assets negatively impacts a group's ability to self-organize (Issac and Walker, 1988).

Group size is another variable that has long been thought to be influence success or failure in natural resource commons as well as software development settings (Schweik et al., 2008). For years it has been thought that the larger the group the higher the coordination costs (Olson, 1965; Brooks, 1975). Yet specifically in open source, "Linus' Law" – "with more eyes, all bugs are shallow" (Raymond, 2001) – suggests that larger groups are actually helpful. Others have found empirical evidence that suggests that the relationship between group size and success is complex, not direct, and probably not linear. For instance, changes in group size tend to simultaneously affect other variables, such as group homogeneity and leadership (Deek and McHugh, 2008: 197). In short, group size has long been thought to be influential, but its relationship in open source commons is unclear.

The last two community attribute variables are *project financing* and *marketing strategies*. Several open source researchers emphasize financing as a key variable for project success (Weinstock and Hissam, 2005; Fogel, 2007). The argument is that financing can ensure that someone is working on the project and provides some assurance that the project will move ahead. At the same time, funding from a particular source could lead to some tensions over future technical direction of the project in the case where there is a hybrid (e.g., volunteer and paid developer) team. Turning to marketing, surprisingly, there appears to be very little in the literature on this as a variable that affects open source success or abandonment. Yet there are indirect suggestions in the literature about the importance of getting the project known in the early days to gain a user community as well as more development support.

2.3 Institutional Attributes of Open Source Commons

The "institutional attribute" category contains variables related to the governance and management systems used by the open source commons and the types of rules in place intended to guide the behavior of participants. Institutions are a key set of variables in natural resource commons used to protect the resource from overuse. However, it is only very recently that researchers are conceptualizing and investigating empirically institutional designs in open source settings (e.g., Schweik and Semenov, 2003; O'Mahony and Ferrarro, 2007; Marcus, 2007; and Schweik and English, 2007). Part of the reason for this lack of attention may be that formal rules and procedures are thought to create disincentives for participating in open source (Raymond, 2001), and in open source commons content version control systems protect the software from accidental destruction through version control and rollback functionality. However given the emerging involvement in open source development by firms, government agencies and nonprofit organizations, it is likely that institutional designs will increasingly be a factor in whether some projects succeed or become abandoned.

To analyze open source institutional designs, we build specifically on the work of Elinor Ostrom (2005) who organizes institutions into Operational, Collective Choice and Constitutional levels. Operational norms and rules oversee the day-to-day activities in a project. Collective choice rules define how changes to operational level rules occur and who has the authority to make such changes. Constitutional level rules specify who is eligible to change Collective Choice rules and also define the procedures for making such changes. They also can be formalized rules that establish the boundaries or principles that the collaboration is grounded upon. The project's open source license is the obvious example of a Constitutional level element. Within each level are seven types of rules (Ostrom, 2005; shown in the "Institutional Attributes" box in Figure 1), but because of space limitations and because Sourceforge.net data does not contain such information, we do not describe this more fully here.

3. An Empirical Analysis of SourceForge.net Projects

We will now give an extremely condensed summary of our recent empirical work related to the variables denoted with an asterisk (*) in Figure 1. Most readers will know that Sourceforge.net (SF) is the largest open source software project hosting site, currently hosting over 130,000 projects. Using available data (FLOSSMole, 2008), along with data we crawled ourselves in the fall of 2006, we compiled a dataset containing of 107,747 SF projects (English and Schweik, 2007). We then organized projects into two longitudinal stages – "Initiation" and "Growth." Projects in the Initiation Stage have not yet produced a first code release. Growth Stage projects have. Next, within these two longitudinal groups, we classified projects as either successful (meaning they continue to be worked on), abandoned or indeterminate collaborations. We then undertook a validation process to verify that the classification system was accurate. See English and Schweik (2007) for more details.

The data we utilize for each SF project consists of five numerical variables and seven "groups" of categorical variables. The five numerical variables include: "Developers," "Tracker Reports," "Page Visits," "Forum Posts" and "Ranking Index." The seven "groups" of categorical variables include: "Intended Audience," "Operating System," "Programming Language," "User Interface," "Database Environment," "Project Topic," and "Project License." Table 1 matches these data to the theoretical concept from Figure 1. It immediately becomes apparent from Figure 1 and Table 1 that the SF data provides measures of some, but not all, potentially influential physical and community attributes, and is quite lean in institutional data. The only institutional characteristic is the open source license used.

Table 1. Selected Variables in Sourceforge.net Data

SF Variable	Description	Associated Theoretical Concept (Figure 1)
Developers	Total number of developers on the project	Group size – Community attribute
Tracker Reports	Total number of bug reports, feature requests, patches and support requests	Collaborative infrastructure – bug tracking system. Physical attribute.
Page Visits	Total number of views of any of the project's SF website	Product utility – Physical attribute; Group size – Community attribute
Forum posts	Total number of Forum posts made to the project's public forums from 2005-10-06 through 2006-08-02	Collaborative infrastructure – Physical attribute; Group size – Community attribute
Downloads	Total number of downloads of the software package	Product utility – Physical attribute; Group size – Community attribute
Intended Audience	Categorical variable describing the type of person the project targets (e.g., end users, advanced end users, business, computer professionals, other)	User Involvement – Community Attribute
Operating System	Categorical variable describing the operating system(s) the software will run on	Product utility, critical infrastructure – Physical attribute
Programming language	Categorical variable(s) describing the programming languages used	Product utility, preferred technologies – Physical Attribute
User Interface	Categorical variable describing how the software interfaces with the user (e.g., command line, GUI, etc.)	Product utility, preferred technologies – Physical Attribute
Database Environment	Categorical variable for the database used in the project's software (if relevant)	Product utility, preferred technologies – Physical Attribute
Project Topic	Group of 19 categorical variables consists of the topics that the SF website uses to classify the projects (e.g., education, games, security, printing, etc.)	Product utility, critical infrastructure – Physical attribute
Project License	Categorical variable(s) describing the type of open source license(s) used.	Constitutional rules – Institutional Attribute

3.1 Statistical Methods: Classification Tree Analysis.

In general, classification techniques include cluster analysis, discriminate analysis, logistic regression, and classification and regression trees. The purpose of these approaches is to efficiently divide the sample data into groups based on one or more independent variables. Classification trees are a unique, nonparametric approach that has several advantages, including accommodation of both categorical and numerical variables, and the ability to model complex interactions (Breiman, et al., 1984). We used classification trees (De'ath and Fabricius, 2000) to test the ability of the SF open source independent variable data to discriminate between projects that were successful and those that were abandoned after they generated a first release of their code – Growth Stage projects.

We initially set out to run classification tree analysis on the entire dataset (n=107,747), but computational requirements were too high. Consequently, we took multiple random subsets to develop Growth Stage trees. Our goal was to determine a representative sample size that would produce useful results, while keeping below the computational threshold. It appears that at n = 1000 or greater, the samples produced instructive and accurate results in most cases. In the results we are about to discuss, we used a random sample of 1000 SF growth stage cases, with categorical variables being assigned a value of 0, 1, or a 2. A value of 0 indicates that the project administrator did not select that independent variable (for example, they do not use the java programming language). A value of 1 indicates that the project administrator did choose that independent variable (e.g., they do use the java language), and a value of 2 indicates that the project administrator did not choose any subcategory of that independent variable group (e.g., they did not answer the Programming Language group entries at all).

3.2 Example of Classification Tree Results.

Given space limitations, we present only one of the classification trees we generated using the SF dataset (Figure 2). As indicated by “cc” percentages, greater than 80% of the projects in the first left and right nodes were correctly classified by dividing the projects by 6,352 page visits. Downloads and Forum Posts further separated successful projects in the right leaves. Moving down the tree on the right side, higher levels of Page Views and use of XWindows (one of the “User Interface” categories) were discriminators of success. Developers and number of downloads contributed to partitioning nodes that contained relatively few observations, and were partitioned with moderate success (cc=0.63 to 0.71). This model correctly classified 80% of the projects, with Kappa statistic = 0.524.

These statistics show that variables that one might expect to be associated with successful projects are indeed associated with success. Page Visits and Downloads are associated with the interest of users in the software and are a measure of product utility (Figure 1). Forum posts are one component of collaborative infrastructure (Figure 1) and indicate an active community where users and developers are communicating. It also suggests a project trying to utilize technology to reduce task granularity by building a question and answer repository that is searchable. Finally, with the exception of the XWindows subcategory of the User Interface group of variables, categorical variables are conspicuously missing from the tree.

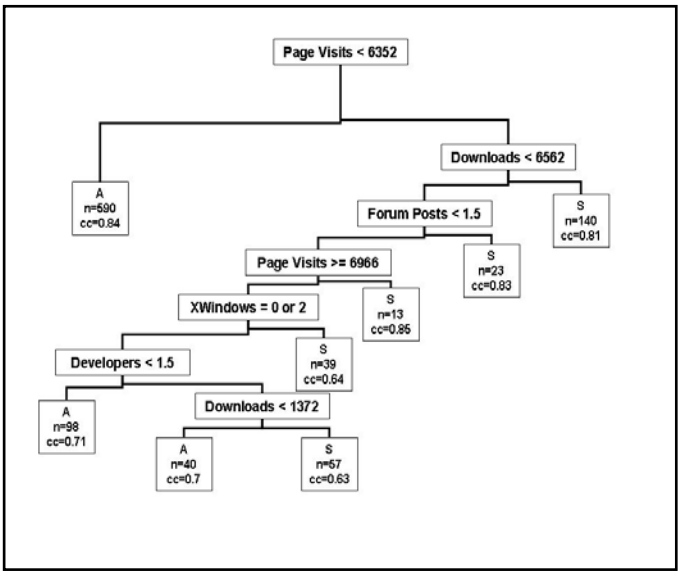

Figure 2. Example of Classification Tree Results Using 1000 Randomly Sampled SF Growth Stage Projects

3.3 Discussion

Classification tree results (including others not shown) suggest that greater software utility (reflected in higher numbers of downloads and page visits) discriminate between successful and abandoned open source projects in the majority of growth stage cases. We intentionally formulated our definition of success to include useful projects having a small number of users, but despite this, having larger numbers of users discriminates between success and abandonment. Also, successful collaborations tend to use the forums and bug tracking features of SF more than the abandoned ones.

We were surprised that our categorical variables (e.g., intended audience, operating system, programming language, database environment, project topics) did not stand out in this or other classification trees. We interpret this result to mean that open source has become a larger, more mainstream phenomenon. In our view, the "user-centric" and volunteer emphasis in past open source literature reflected, at least in part, programmers building software that they needed to support the continued buildup of open source technologies (e.g., Linux and related software, web and email processing, etc.). The fact that none of the categories are important discriminators in our data suggests that people are collaborating in all kinds of open source projects; that is, success and abandonment are widely distributed across all types of software projects rather than being confined to fewer categories as they may have been in the past. This analysis emphasizes the importance of community attributes over physical attributes in explaining success or abandonment of open source commons. Finally, except for the finding that GPL versus non-GPL licensing is not an important distinguishing variable, the role of institutional attributes still remain an open question given SF does not contain that kind of data.

4. Conclusions

In this paper we presented a multidisciplinary literature review summarizing factors thought to influence collaborative success in open source commons (Figure 1). Where possible, we then matched up the theoretical factors identified in the literature review to project data found on Sourceforge.net (Table 1). We then undertook classification tree analysis to investigate whether certain factors discriminated between successful and abandoned SF projects with at least one public release (Figure 2). Our results suggest that product utility and group size (specifically a user base) are important in ensuring successful ongoing open source development collaborations. Moreover, the fact that none of the categorical variables stand out reveals that success and abandonment of open source projects occurs in all varieties of software. Finally, our analysis shows that the SF project database provides only limited utility in understanding success and abandonment of open source projects. Many of the potentially important community and institutional variables are not available. More empirical research is needed to investigate these other potentially influential factors.

Acknowledgments

Support for this study was provided by grants from the U.S. National Science Foundation (NSFIIS 0447623 & 0630239). Any opinions, findings, conclusions, or recommendations expressed in this material are the authors and do not necessarily reflect the views of the National Science Foundation.

References

Breiman, L, Friedman, JH, Olshen, RA, and Stone CG, 1984, *Classification and Regression Trees,* Wadsworth International Group, Belmont, California.

Brooks FP Jr. [1975] 1995, *The Mythical Man-Month: Essays on Software Engineering,* Anniversary Edition, Addison-Wesley, Reading, MA.

De'ath, G & Fabricius, KE, 2000, 'Classification and regression trees: a powerful yet simple technique for ecological data analysis', *Ecology*, vol 81, no. 11, pp. 3178-3192.

Deek, FP and McHugh JAM, 2008, *Open Source Technology and Policy*, Cambridge University Press, New York, NY.

English, R & Schweik, CM, 2007, 'Identifying success and abandonment of free/libre and open source (FLOSS) commons: a preliminary classification of sourceforge.net projects', *Upgrade: The European Journal for the Informatics Professional*, vol VIII, no. 6, viewed 20 July 2008, <http://www.upgrade-cepis.com/issues/2007/6/upg8-6English_Schweik_v2.pdf>.

Ewusi-Mensah, K, 2003, *Software development failures,* MIT Press, Cambridge, MA.

FLOSSMole, 2008, viewed July 22, 2008, <http://sourceforge.net/projects/ossmole/>.

Fogel K, 2007, *Producing Open Source Software: How to Run a Successful Free Software Project,* O'Reilly Media, Sebastopol, CA. Viewed 13 July 2008, <http://producingoss.com/>.

Hess, C, and Ostrom, E (eds.), 2007, *Understanding Knowledge as a Commons: From Theory to Practice,* Cambridge, MA, MIT Press.

Issac, MR & Walker J, 1998, 'Group size effects in public goods provision: the voluntary contribution mechanism', *Quarterly Journal of Economics*, vol 53, pp. 179-200.

Katzenbach, J & Smith D 1993, 'The discipline of teams', *Harvard Business Review*, vol 71, pp. 111-120.

Markus ML, 2007, 'The governance of free/open source software projects: monolithic, multidimensional, or configurational?', *Journal of Management and Governance*, vol. 11, no. 2, pp. 151-163.

Nardi, BA & Wittaker S 2002, 'The place of face to face communication in distributed work', in P Hinds, and S Kiesler (eds.), *Distributed Work*, MIT Press, Cambridge, MA.

O'Mahony, S & Ferraro F 2007, 'Emergence of governance in an open source community', *Academy of Management Journal*, vol. 50, no. 5, pp. 1079-1106.

Olson M, 1965, *The Logic of Collective Action*. Harvard University Press, Cambridge, MA.

Ostrom E, 2005, *Understanding Institutional Diversity*, Princeton University Press, Princeton, N.J.

Ostrom E, Burger J, Field CB, Norgaard RB, & Policansky D 1999, 'Revisiting the commons: local lessons, global challenges', *Science*, vol. 284, pp. 278-282.

Putnam RD 2007, 'E pluribus unum: diversity and community in the twenty-first century. The 2006 Johan Skytte prize lecture', *Scandinavian Political Studies*, vol. 30, no. 2, pp. 137-174.

Raymond E, 2001, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Sebastopol, CA.

Sandler T, 2004, *Global Collective Action*. Cambridge University Press, Cambridge MA.

Schweik CM.& Semenov A 2003, 'The institutional design of 'open source' programming: implications for addressing complex public policy and management problems', *First Monday*, vol. 8, no. 1, viewed July 23, 2008, <http://www.firstmonday.org/issues/issue8_1/schweik/>.

Schweik CM & English R 2007, 'Conceptualizing the institutional designs of free/libre and open source software projects', *First Monday*, vol. 12, no. 2, viewed 23 July, 2008, <http://www.firstmonday.org/issues/issue12_2/schweik/index.html>.

Schweik CM, English R, Kitsing, M. & Haire, S 2008, 'Brooks' versus Linus' law: an empirical test of open source projects', in SA Chun, M Janssen, and R Gil Garcia (eds.), *The Proceedings of 9th International Digital Government Research Conference*, Montreal, Canada.

Tyran KL, Tyran, CK, & Shepherd M 2003, 'Exploring emerging leadership in virtual teams', In CB Gibson, and SG Cohen (eds.), *Virtual Teams that Work: Creating Conditions for Virtual Team Effectiveness*. Jossey-Bass, San Francisco, CA.

Varughese G & Ostrom E 2001, 'The contested role of heterogeneity in collective action: some evidence from community forestry in Nepal', *World Development*, vol. 29, no. 5, pp. 747-765.

von Hippel E & von Krogh G 2003, 'Open source software and the 'private-collective' innovation model: issues for organization science', *Organization Science*, vol. 14, no. 2, pp. 209-223.

von Hippel E, 2005, *Democratizing Innovation*. MIT Press, Cambridge, MA.

Weinstock, CB & Hissam, SA 2005, 'Making lightning strike twice', In J Feller, B Fitzgerald, SA Hissam and K R Lakhani (eds.), Perspectives on Free and Open Source Software. The MIT Press, Cambridge, MA.