Understanding Open Source Communities as Complex Adaptive Systems: A Case of the R Project Community

Full Paper

Georg J.P. Link
University of Nebraska at Omaha
glink@unomaha.edu

Matt Germonprez
University of Nebraska at Omaha
mgermonprez@unomaha.edu

Abstract

Open source communities evolve. This evolution is, at times, driven by corporate engagement with those communities. In these corporate-communal contexts, open source foundations often serve as facilitators in the evolution process and make these arrangements more stable over time. This paper expands the application of complex adaptive systems (CAS) theory to understand the role of open source foundations as facilitators in the evolution of corporate-communal arrangements. We present the case of the R Project community and how we can leverage complex adaptive systems as a way to understand the evolution of the community as driven by corporate engagement and facilitated by open source foundations. We develop the theory of CAS by enhancing the understanding of attractors in the evolution of CAS.

Keywords

Open Source Community, Open Source Foundations, Corporate Engagement, Complex Adaptive Systems.

Introduction

Open source communities evolve (Crowston et al. 2012) as they grow from infant communities of a few collaborative members sharing source code, to highly active communities, comprised of thousands of members developing source code that shapes the world we live in. A key influence often driving this evolution is increasing corporate engagement (Fitzgerald 2006), as the open source movement outgrows its roots from the free software movement (Kelty 2008). Research has found that corporations might choose an open source model to co-innovate non-differentiating technology with rivaling competitors (Germonprez et al. 2013a) because the advantages of using open source for innovation often outstrip the costs of having to maintain internal intellectual property (Hippel and Krogh 2003).

In this setting, there are several evolutionary issues when corporations engage with open source communities, as the influence of corporations on the community needs to be balanced with the interests of the entire community (Dahlander and Magnusson 2005). We know that there are no contractual means to enforce compliance with community rules (Feller et al. 2008), sometimes resulting in a strong presence of corporations in open source communities which can deter volunteers or other corporations from contributing (Spaeth et al. 2015). However, such issues can be negotiated in an evolving community. These issues determine the interaction between members and thereby the future of a community.

A well-tested method to deal with these issues is to engage a neutral third-party facilitator. Such a facilitator supports community processes, helps establish best practices, and provides a fair environment where everyone can participate as equals (Germonprez et al. 2013a). To this end, open source communities established not-for-profit foundations, such as the Linux Foundation, Apache Software Foundation, Eclipse Foundation, and the R Foundation. Through the affiliation with a foundation, the evolution of an open source community is influenced and shaped differently (Beecher et al. 2009). Foundations satisfy the need for stable and reliable open source communities and software, fueled by corporations who integrate open source in their product development activities and rely on open source communities for long-term and stable sources of innovation (Germonprez et al. 2013c).

In this paper we explore the R Project community, as it illustrates the facilitating role of a foundation in guiding the evolution as corporate engagement increases. "R is a system for statistical computation and graphics" (Hornik 2015), and over the past few years, R was widely adopted in academia and industry. In 2014, corporations met with the R Community to find a way to financially support and advance the software, the community, and its infrastructure (Plummer 2015). The R Foundation played an important role in this discussion. In 2015, the R Consortium was established, stemming from discussions with the Linux Foundation. Through these discussions, The R Consortium became a Collaborative Project at the Linux Foundation and thus benefits from the know-how and experience of another leading open source foundation. In November 2015, the R Consortium announced funding of its first project, R-Hub, with the aim of improving the development process (Machlis 2015). Strikingly, there was little response from the community toward this financial investment, unlike other documented cases where communities responded negatively to the investment and consequential changes in corporate-communal engagements (Germonprez and Hovorka 2013b). This leads to our research question, considering the precise role of foundations to broker the complexities that stem from corporate-communal engagements:

> *How can the role of open source foundations and corporate engagement be understood in the evolution of open source communities?*

In response, we explore foundational-corporate-communal engagements as complex adaptive systems (CAS), providing a framework within which to understand their evolution. CAS allows understanding of the roles of open source foundation and corporations in the processes of communal evolution. We make two primary contributions in this paper. First, we contribute to CAS theory in the context of open source communities that are affiliated with open source foundations. We develop CAS theory by conceptualizing open source foundations as attractors that guide the evolution of an open source community, and by observing that the creation of a new attractor has been driven by corporate engagement. Second, we contribute to practices within open source communities, explicating how to make informed decisions when affiliating with a foundation. We help identify how to better understand foundational roles within open source communities and the impact that this can have on communal evolution.

Open Source Communities

As open source communities evolve, early stages may not require an open source foundation for success. Many open source communities were started by one or a few individuals who had "an itch to scratch" or a concrete problem to solve (Raymond 2000). The typical progression is that following a public announcement by the project initiators about their initial code, other people show interest and start contributing. Contributions include bug reports, patches that fix bugs, feature request, patches to implement new features, support questions, and answers to these questions. As the community grows and more people interact, the community founders often see the need to share the load of maintaining the software. Early research on open source often focused on these internal processes that were faced by open source communities (Aksulu and Wade 2010).

Recently, corporations have started to engage with open source communities for economic, social, and technical reasons (Dahlander and Magnusson 2005). This can include a gain in competitiveness, cutting costs, sharing code with community, exploiting feedback from the community, reaching higher diffusion of innovation, winning adoption, and promoting standards. When choosing the reasons for engagement, corporations may use freely available open source software but ignore community norms, values, or rules (parasitic), respect the community, or maybe even engage with it, but not contribute back (commensalistic) (Dahlander and Magnusson 2005).

Extractive perspectives do not represent the engaged corporate-communal relationship that open source communities now often represent (Dahlander and Magnusson 2005). A symbiotic relationship is evident as corporations engage with open source communities, often paying employees to participate for reasons of both corporate and communal interest. Employees bring value to a community by having access to corporate resources, being able to respond and act faster, and by having high levels of interest in participating (Dahlander and Wallin 2006). Through the high level of interaction, employees often become central members of a community and influence the evolution of it.

Finally, a community may come to need an open source foundation to facilitate its continued success and long-term stability (Riehle 2010). "The main purpose of a foundation is to act as the steward of the software

being developed and to ensure its long-term survival" (Riehle 2010, pp. 86-87). A community is not itself a legal entity that can receive donations, possess assets, or voice a shared opinion. Community members often come to realize that doing business as a collection of individuals has limits and may thus require foundation support. As a supporting organization, a foundation assumes several responsibilities, including organizing the open source community, actively marketing the community, clarifying and managing intellectual property, setting strategic directions for the community, responding and remaining accountable to its members, and running all relevant back-office processes (Riehle 2010). In response to some or all of these needs, an open source community may either partner with an existing foundation (e.g., OpenOffice and the Apache Foundation or FOSSology and the Linux Foundation), or start a new foundation (e.g., R Project founded the R Foundation or LibreOffice founded the Document Foundation) as a way to legally structure that community. These different research perspectives on open source communities, corporations, and foundations are depicted in Figure 1.

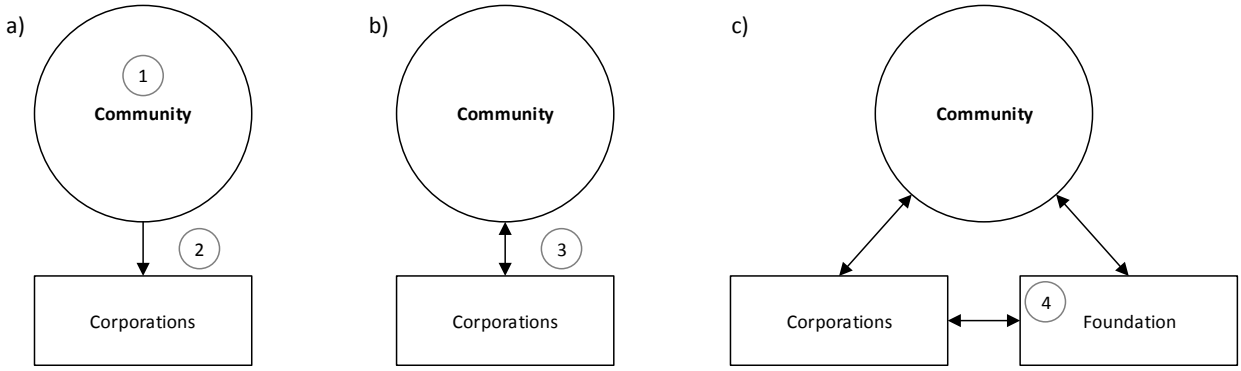

Figure 1. Early research on open source communities often involved the examination of the dynamics of the community itself (1). Later research explored the engagement from corporations, but much of this research presented the community as a source of innovation resources for corporations (2). We understand that corporate-communal engagements were mutual partnerships with the intention of benefiting both the corporation and the community (3). Finally, we understand that foundations can play an instrumental role in stabilizing and facilitating the complexities associated with corporate-communal engagements (4).

In this research, we explore the role of foundations as a stabilizing environment for open source communities, providing a trusted and neutral environment for stable and explicit communal structures. We explore the increased neutrality and stability that foundations provide as they manage an open source community, prevent too strong a corporate influence, and provide a level playing field for everyone.

In an open source community, members participate voluntarily and can easily leave at any time, an important consideration for corporations that plan to engage with a community. Just as we have seen that the affiliation of open source projects with corporations can have detrimental effects on community perception and engagement (Spaeth et al. 2015), corporate influence and overreaching can alienate a community and even drive away members (Germonprez and Hovorka 2013b). By stabilizing the community with the help of a foundation, corporations gain trust in the reliability and neutrality of an open source community as part of their innovation practices, and vice versa (Levy and Germonprez 2015).

Research on open source communities has focused on different aspects, starting with the internal communal processes, then including corporate engagement, and recently including the role of foundations. This shift recognizes the many members in an open source community. To understand how relationships between community members form and evolve, CAS theory provides the necessary conceptualization. In doing so, we come to understand open source engagements as being guided by open source foundations.

Theoretical Foundation: Complex Adaptive Systems

Complex adaptive systems emerge from the interaction of connected members with each other (Anderson 1999). CAS provides a theory to model complex behavior and understand how communities evolve, as understood through four elements: (i) members with decision rules, (ii) self-organizing networks, (iii) co-

evolution of members, and (iv) system evolution (Anderson 1999). CAS theory has been applied to open source communities before (Agnihotri et al. 2012; Muffatto and Faldani 2003), and we build from this work and expand the understanding of CAS theory.

Members of open source communities are users, developers, and corporate and foundation employees (Muffatto and Faldani 2003). To consider evolution of an open source community, one assumes that the current state was influenced by individual members. The decision of individual members about which technologies to use for communication will shape the future communication patterns of the community. Likewise, decisions such as which members get to commit code to the main repository shape the structure of the community. The behavior of each member is determined by decision rules (Anderson 1999), and the decision rules may be different among members and may evolve over time. For example, the choice to join a community might depend upon how a potential new user observes the mailing lists to be active. Often, members aim to increase personal gains and base communally-impactful decisions on these goals.

Within open source communities, members are only partially connected and may act on different levels or within different sub-groups. Core-maintainers might only interact with a select set of trusted community members who might be responsible for different aspects of the project. These trusted community members might interact with other members who are contributing to their particular aspect of the project. The behavior of individuals is influenced by local information, but yet the whole community remains globally organized. Patterns of interaction tend to emerge naturally throughout different community locations. To maintain a state of order, the community relies on continued contributions, and it is from this participation that the structure in a community emerges.

While every community member adapts to their environment in the pursuit of increasing personal gains, the potential gains depend on the current state of an ever-changing environment. This dynamic explains the constant need for adjusting behavior, and since members influence each other, the dynamic in an open source community is always in flux. The behaviors are not entirely random but may appear from an outside perspective as chaotic nonetheless.

Complex adaptive systems evolve over time because community members are joining, leaving, or evolving themselves. Furthermore, the interaction between members is likely to change as some members take on new responsibilities, some choose to contribute in different ways, and others reduce their involvement. It is important to realize that the community is interconnected and that everything depends on other parts and the environment.

Further, a CAS, such as an open source community, evolves around attractors (Anderson 1999) who impact possible developments. Attractors are well known in physics where a geometrical understanding states that, over time, a complex system might stop exploring the available space and follows an attractor (Eckmann and Ruelle 1985). See Figure 2a for an example. Organizational adoption of CAS illustrates that "attractor states" can be an equilibrium because networks tend to return to an equilibrium state as the system evolves (Anderson 1999). Figure 2b illustrates that the attractors can be distinct from the attractor state. In open source communities, we conjecture that an attractor could be an open source foundation, which through its supporting role makes decisions on behalf of the community. The attractor provides guidance to the community and because of an attractor's status and role in the community, it is unlikely that these decisions will be overturned.

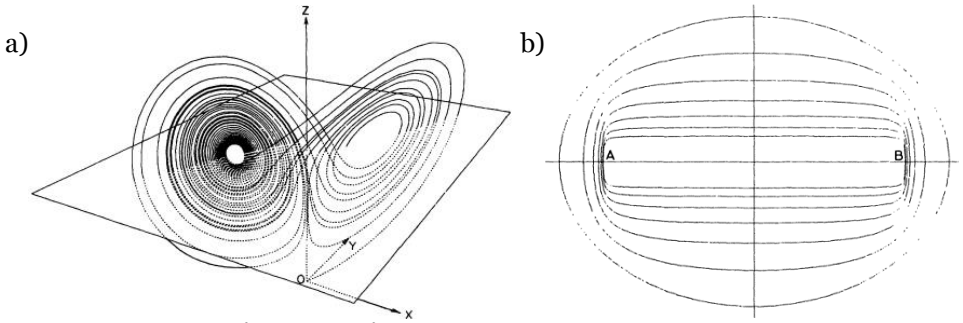

Figure 2. a) A system's evolution over time revolves around the attractor. b) The whole space is attracted to the segment A,B but only A and B are attractors. (Images from Eckmann and Ruelle 1985, p. 623).

To our knowledge, attractors in CAS have not been studied from an open source community perspective. Unlike physical systems, open source communities cannot be expressed in mathematical functions. The next section describes how we researched attractors in open source communities.

Method

Our methodological approach is netnography (Kozinets 2015), an ethnographic approach adjusted to online environments. The objective of Netnography is to learn and understand online communities, and since open source communities largely collaborate online, this method is appropriate. A fundamental idea is for the researcher to fully immerse in the community and learn by being part of the community. The study of evolving communities "does not always afford the luxury of planning a research agenda, entering a prechosen community or even allowing a return to the field as needed" (Germonprez and Hovorka 2013b, p. 529). Netnography often requires a sense of immediacy as researchers are presented with fast-moving, short timeframe events that nonetheless carry significant research potential.

To represent our netnography, we utilize impressionist tales (Van Maanen 1988). Impressionist tales are the creation of repesentations, the process necessary to illustrate observations. The reader of impressionist tales is abstracted from the actual events and is reliant on the researcher to provide a representive view into the situations. It is important to note that "an impressionist tale is not defined by the amount of time spent or the volume of data collected regarding a community, and time/volume should not serve as a proxy measure for the quality of the data, engagement with a social network, or communal understanding of a social network" (Germonprez and Hovorka 2013b, p. 531). Rather, the purpose of an impressionist tale is to tell a rich story that connects the reader to the observed community. We tell the tale of the R Project community through an impressionist tale to understand the role of open source foundations and corporate enagement in the evolution of the community. We also contribute to theory by developing an understanding of attractors in the evolution of CAS.

Research Context

Our netnographic context is the R Project community, and specifically the development of the R Consortium. The R Project originated as a software project for teaching statistics at the university level in 1993. Interest from other people was great, which motivated the original authors to open source the software in 1995. By 1997, a core group was established to share the responsibilities of developing and maintaining the software. The R Foundation was founded six years later in 2003 as an Austrian non-profit organization. The R Foundation holds and administers the copyright on the source code. Corporations and universities adopted R for performing data analyses (Vance 2009). In the market for advanced analytics software, R is among the top three software packages besides commercial vendors SAS and SPSS (R. A. Muenchen 2015).

One of the success factors of R is the ability to develop packages that extend the functionality of R and the ability to share these packages with the community. As of this writing, "the CRAN package repository features 8,015 available packages" (R-Project 2016). Of those, 2,762 packages were updated in the six months from September 2015 to February 2016. This illustrates an active user base of the R Community. As a way to support these activities, US-based companies wanted a non-profit foundation where they could channel their contributions to the R Project. After discussions with the R Foundation, the R Consortium was established in 2015 with the mission to "support the R community, the R Foundation and organizations" ("R Consortium" 2016).

We began our investigation a few months after the R Consortium had announced its first major project. We treated the internet as a data archive, since "online interactions often leave a more permanent trace" (Kozinets 2015, p. 74). Our data sources included all official mailing archives (i.e., R-announce, R-help, R-package-devel, R-devel, and R-packages), twitter feeds, websites and blogs, and GitHub repositories assocated with the project. We searched within these data sources for "R Consortium", "R-Hub" (the name of the first funded project), the name of the person who was assigned to implement R-Hub, "Linux Foundation" that hosts the R Consortium, and variations like "consorti*" and "hub" in case they were used in abbreviated forms.

Findings

The Shifting Attractors of the R Project Community

We begin the impressionist tale at the genesis of the R Project. Ross Ihaka and Robert Gentleman, both from the University of Auckland, needed a better software environment for teaching statistical computing (Ihaka 1998). They developed a software tool with the mindset of statisticians. In August 1993, they made a small announcement on the s-news mailing list about their implementation of R and linked to a binary version. Among the interested people was Martin Mächler of ETH Zurich, who convinced Ihaka and Gentleman to release the R source code under the open source license GPL (Ihaka 1998). His engagement shaped the evolution of the R Project community, especially by hosting to this day the official mailing lists on the ETH Zurich domain. The decision of the initiators Ihaka and Gentlemen to share their work started an evolution of the R Project community. As the first attractors, these three shaped the early evolution of the R Project community through their decisions.

A network of collaborating members then formed, which was in need of adequate communication channels. Following the release in June 1995, the community used automated mailing lists to enable increasing contributions. At first, only an R-testers mailing list was started. Later, it was replaced by R-announce, R-help, and R-devel, which are still in use today. The use of multiple mailing lists organizes the collaboration into sub-groups. By mid-1997, the community exhibited more self-organization when the "core group" of maintainers (today called the R Development Core Team) emerged to share the load of the development (Ihaka 1998). The initiators no longer served as attractors in the evolution process, since they reduced their influence by sharing responsibilities and essentially passing them on to the developing core group. The community began to evolve around the core team as the new attractor.

The R Project community grew and evolved in the following years. By 2003, the community established a foundation. The R Foundation was founded and announced in April 2003 (Gentleman et al. 2003). While the R Development Core Team remained the attractor for evolution in regard to the development of the R software, it created the R Foundation as the new attractor to shape community evolution. The R Foundation was "founded by the members of the R Development Team in order to:

- Provide support for the R project and other innovations in statistical computing. We believe that R has become a mature and valuable tool and we would like to ensure its continued development and the development of future innovations in software for statistical and computational research.
- Provide a reference point for individuals, institutions or commercial enterprises that want to support or interact with the R development community.
- Hold and administer the copyright of R software and documentation." (The R Foundation 2016)

Corporate engagement also grew at this time. Corporations integrated R into organizational processes and offered services and products based on R as they interacted more closely with the community. The R Foundation provided guidance for this evolution and aligned the driving force of corporate engagement with community interests.

The Interplay of Attractors

On November 2, 2015, the R Consortium funded its first project (Machlis 2015). The project is called R-Hub and received a grant of $85,000. The goal for R-Hub is to "modernize and improve the entire process of developing and testing R packages" (Machlis 2015). R packages are community developed add-ons that enhance and add functions to the R core software. Every day, about 30 packages are updated, which can be a laborious process. R-Hub would allow developers to pre-test packages and thus reduce the work load during the CRAN submission and testing process. The service will be compatible with existing infrastructure and available for free to all package developers.

We expected that the announcement would elicit a reaction from the R Community because it affects an important communal process. Following the press release, it was re-tweeted, and blog posts picked up on the story. On Twitter, almost all tweets referenced a news article or blog post immediately after the press release. However, no discussion occurred from these announcements, and after a few days the mentions faded quickly. Further, we did not observe comments or other reactions, not even at the time of the press release.

Searching through the official mailing lists R-announce, R-help, R-package-devel, R-devel, and R-packages, almost no reference was made to the R-Hub project. The R-Hub project was only mentioned once, on Jan 18, 2016 in a reply on R-devel, as a possible solution to a problem someone asked about. This does show that some community members were aware of the project. However, the lack of response is telling. We explored this lack of interest by exploring the code repositories for discussion. Indeed, we found seven issues on GitHub commenting on and asking about the project proposal. The number of issues concerning the code development that were from someone other than the developer was essentially nonexistent (one issue was found). Considering the prospective attractive value that this project can have for the R Community and the amount of funding that is provided, we expected to see more engagement, especially since R-Hub could serve as an attractor to drive evolution of the highly active processes around developing and updating packages.

To understand this lack of reaction, we had to look beyond the public discussion and consider structural elements that could play a role. We know that companies based in the US were looking for a way to make financial contributions to the R Community and support its infrastructure (Plummer 2015). However, the R Foundation that manages the assets of the R Community and accepts donations is located in Vienna, Austria. This did not allow US-based companies to make tax-deductible contributions.

To address this problem, at DSC 2014: Directions in Statistical Computing, US-based corporations met with the R Foundation to discuss an R Consortium. In the following months, the R Foundation, in its function as the attractor for community evolution, approved the idea and worked with the corporations and the Linux Foundation in setting up the R Consortium. It was set up as a trade association that was recognized as a non-profit organization under the US Internal Revenue Service code 501(c)(6). The R Consortium was also established as a Collaborative Project with the Linux Foundation. This means that the R Consortium is now guided by the Linux Foundation in establishing its processes, how to engage with the open source R Community, and other best practices. As a collaborative project at the Linux Foundation, members of the R Consortium are primarily commercial corporations that have a status with the consortium depending on their level of financial contributions to the R Consortium. The only exception is the R Foundation, which has a special and permanent membership without financial obligations. The R Foundation, as the permanent member, can advise and guide the actions of the R Consortium to be aligned with the R Community's interests. The goal of the R Consortium is to "support the R Community, the R Foundation and organizations developing, maintaining & distributing R software" (www.r-consortium.org) but to not influence with core software development of R itself, which remains under the control of the R Community. The R Consortium aspires to be an attractor for corporate engagement that is guided by the R Foundation as the attractor for community evolution.

On June 30, 2015, one year after corporations sought guidance from the R Foundation on how to best be engaged, the R Consortium was announced at the annual international R user conference, useR! (Linux Foundation 2015). The responses on Twitter were positive.

- "Good day for R:Linux Foundation Announces R Consortium to Support Millions of Users Around the World <link: press release>" @ralf_mueller (1 Jul 2015)
- "Open source lives! The R project is the real deal: In announcing the R Consortium to work in concert with the… <link: blog post>" @GetItAl (1 Jul 2015)
- "Very excited about launch of R Consortium: high expectations from such a team <link: press release>" @BugsWormsNBats (2 Jul 2015)
- "I accept and embrace our new overlords. 'The R Consortium: what does it mean for me?' <link: blog post>" @convalytics (2 Jul 2015)

Beyond Twitter, little other response was observed online. Most blog posts or news articles were very similar to the press release. An exception was a blog post titled "Goals for the New R Consortium" (B. Muenchen 2015), which made suggestions for issues the R Consortium could address. On the official mailing lists, there is no announcement related to the R Consortium and the Linux Foundation. No emails picked up the announcement, and no community member voiced an opinion or started a discussion on the mailing lists. Two emails referenced the R Consortium, but were not related to the first announcement. One is an email reply on the R-package-devel list on Jul 21, 2015, in which the author hints that a previously proposed idea might be interesting for the R Consortium. The second email is an announcement on R-help, on Nov 4,

2015, where the R Consortium calls for project proposals. This email was sent several months after the announcement of the creation of the R Consortium. Interestingly, this email received no replies on the mailing lists. To date, it is not evident that the addition of the new attractors to the R Community lead to a sudden change in the evolution of the community.

Discussion and Conclusion

The R Community is a complex adaptive system influenced by numerous attractors and as the community evolved and grew the attractors changed. The core team reduced its influence but remained the attractor for software evolution when it established the R Foundation as a new attractor for community evolution. The introduction of the R Foundation was an evolution of the community in response to the legal environment and to provide a home for the R Project community. The driving corporate engagement led to the establishment of the R Consortium as a new attractor for corporate engagement. The attractor structure of the R Project community as it evolved through the engagement of all members is depicted in Figure 3.

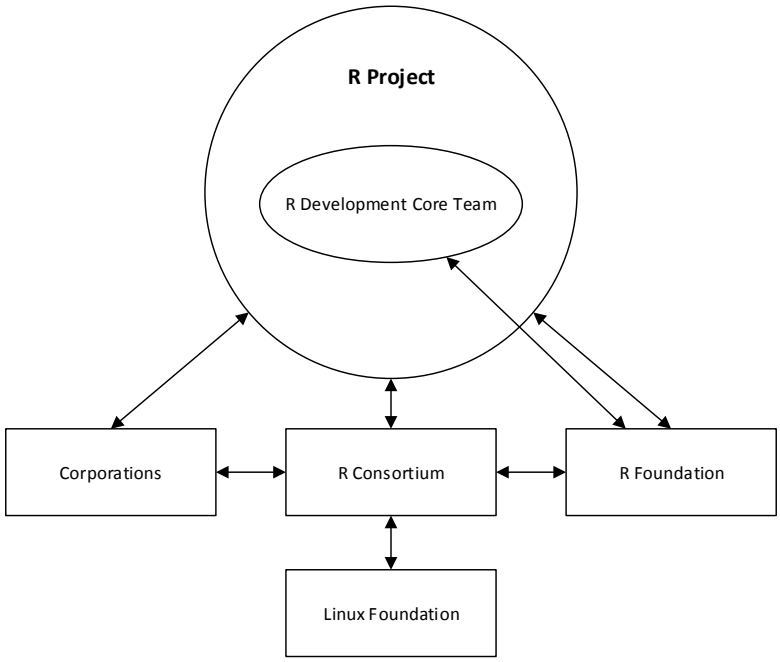

Figure 3. The R Project Community. At the core is the R Development Core Team, the R Foundation is part of the community but oversees the evolution. Corporations engage directly with the R Project community and through the newly founded the R Consortium that is guided by the R Foundation and hosted by the Linux Foundation.

In observing this, we contribute to theory by adapting CAS theory to the context of the open source R Project and its evolution. We extend the understanding of attractors in open source communities guiding forces in the evolution. Attractors can be members or governance bodies as communities organize around these attractors throughout their evolution.

A novel observation is that attractors change as the system evolves. We observed attractors voluntarily reduce their influence on the community evolution to allow for more growth and attractors share influence by introducing new attractors. We also observed that attractors in CAS guide different evolutions like the R Development Core Team guides software evolution, the R Foundation guides community evolution, and the R Consortium guides the evolution of corporate engagement.

Finally, we contribute to practice by reporting the preliminary story of aligning the driving corporate engagement with an open source community through engagement with open source foundations. Open source communities face decisions about how to manage the ongoing evolution within the community and

of its members. At some point the prospect of joining a foundation might seem appealing. This research provides communities with a way of understanding the complexities such that corporations and open source foundations can adjust their understanding in the evolution of open source communities and adjust their behavior accordingly.

Limitations and Directions for Future Research

A key limitation to this study is that the authors are not active participants in the R Project community. A community member might have background knowledge that makes the stated facts appear in a different light. Further, data collection started after R Consortium was established. Tacit and situation-sensitive information could not be considered in this report since they do not get logged in public archives. Future research will strengthen findings with interviews and experience reports from community members who were part of the evolution. Drawing conclusions about the evolution caused by the R-Hub project of the R Consortium is limited since it is in an early phase. The case will be explored over the next several years to more deeply understand community evolution and roles that different attractors play in this process.

Future studies can further develop theory about attractors in organizations and communities. Open questions include: How can we identify attractors? What are preconditions for new attractors to emerge and how do they form? What is the impact reach of attractors, especially as communities grow? How many attractors can influence the same community at one time? What causes attractors to change?

Finally, we chose CAS theory as our theoretical lens and a different theoretical lens will likely lead to different interpretations of the data. Future research can evaluate our findings by choosing a different theoretical lens; for example, Motivation Perspective, Social Identity Theory, Public Goods Theory, Social Network Perspective, or Organizational Perspective (Agnihotri et al. 2012). Each community is different and small differences in the environment and ways of looking at these environments can lead to unique future outcomes.

REFERENCES

Agnihotri, R., Shanker, M., and Kothandaraman, P. 2012. "Theorization of the Open Source Software Phenomenon: A Complex Adaptive System Approach," *Journal of Management and Marketing Research* (10), Academic and Business Research Institute (AABRI), p. 1.

Aksulu, A., and Wade, M. 2010. "A Comprehensive Review and Synthesis of Open Source Research," (11:11) (available at http://aisel.aisnet.org/jais/vol11/iss11/6/; retrieved March 1, 2016).

Anderson, P. 1999. "Complexity Theory and Organization Science," *Organization Science* (10:3), pp. 216–232.

Beecher, K., Capiluppi, A., and Boldyreff, C. 2009. "Identifying Exogenous Drivers and Evolutionary Stages in FLOSS Projects," *Journal of Systems and Software* (82:5), pp. 739–750 (doi: 10.1016/j.jss.2008.10.026).

Crowston, K., Wei, K., Howison, J., and Wiggins, A. 2012. "Free/Libre Open-source Software Development: What We Know and What We Do Not Know," *ACM Computing Surveys* (44:2), New York, NY, USA: ACM, pp. 7:1–7:35 (doi: 10.1145/2089125.2089127).

Dahlander, L., and Magnusson, M. G. 2005. "Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms," *Research Policy* (34:4), Elsevier BV, pp. 481–493 (doi: 10.1016/j.respol.2005.02.003).

Dahlander, L., and Wallin, M. W. 2006. "A Man on the Inside: Unlocking Communities as Complementary Assets," Research Policy (35:8), pp. 1243–1259 (doi: 10.1016/j.respol.2006.09.011).

Eckmann, J.-P., and Ruelle, D. 1985. "Ergodic Theory of Chaos and Strange Attractors," *Reviews of Modern Physics* (57:3), p. 617.

Feller, J., Finnegan, P., Fitzgerald, B., and Hayes, J. 2008. "From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks," *Information Systems Research* (19:4), pp. 475–493.

Fitzgerald, B. 2006. "The Transformation of Open Source Software," *MIS Quarterly* (30:3), pp. 587–598.

Gentleman, R., Ihaka, R., and Leisch, F. 2003. "E-Mail: R Foundation for Statistical Computing," April (available at https://stat.ethz.ch/pipermail/r-announce/2003/000385.html; retrieved 3/1/2016).

Germonprez, M., Allen, J. P., Warner, B., Hill, J., and McClements, G. 2013a. "Open Source Communities of Competitors," ACM I*nteractions* (20:6), Association for Computing Machinery (ACM), pp. 54–59 (doi: 10.1145/2527191).

Germonprez, M., and Hovorka, D. S. 2013b. "Member Engagement Within Digitally Enabled Social Network Communities: New Methodological Considerations," *Information Systems Journal* (23:6), pp. 525–549 (doi: 10.1111/isj.12021).

Germonprez, M., Kendall, K., Kendall, J., Young, B., Warner, B., and Mathiassen, L. 2013c. "The Domestication of Open Source," in *DIGIT 2013 Proceedings*.

Hippel, E. von, and Krogh, G. von. 2003. "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science," *Organization Science* (14:2), pp. 209–223.

Hornik, K. 2015. "R FAQ," (available at https://CRAN.R-project.org/doc/FAQ/R-FAQ.html; retrieved March 1, 2016).

Ihaka, R. 1998. "R: Past and Future History," *Computing Science and Statistics*, pp. 392–396.

Kelty, C. M. 2008. *Two Bits: The Cultural Significance of Free Software*, Durham: Duke University Press.

Kozinets, R. V. 2015. *Netnography: Redefined*, SAGE Publications.

Levy, M., and Germonprez, M. 2015. "Is it Egalitarianism or Enterprise Strategy? Exploring a New Method of Innovation in Open Source," (available at http://aisel.aisnet.org/amcis2015/StrategicUse/GeneralPresentations/18/; retrieved March 1, 2016).

Linux Foundation. 2015. "Linux Foundation Announces R Consortium to Support Millions of Users Around the World,", June (available at http://www.linuxfoundation.org/news-media /announcements/2015/06/linux-foundation-announces-r-consortium-support-millions-users; retrieved March 1, 2016).

Van Maanen, J. 1988. *Tales of the Field: On Writing Ethnography*, The University of Chicago Press.

Machlis, S. 2015. "New 85K Package Development Service Backed by R Consortium", November (available at http://www.computerworld.com/article/2999797/application-development/r-consortium-funds -new-r-package-development-service.html; retrieved March 1, 2016).

Muenchen, B. 2015. "Goals for the New R Consortium", July (available at http://r4stats.com/2015/07/28/goals-for-the-new-r-consortium/; retrieved March 1, 2016).

Muenchen, R. A. 2015. "The Popularity of Data Analysis Software," (available at http://r4stats.com/popularity; retrieved March 1, 2016).

Muffatto, M., and Faldani, M. 2003. "Open source as a Complex Adaptive System," *Emergence* (5:3), Taylor & Francis, pp. 83–100.

Plummer, M. 2015. "The R Consortium and the R Foundation," *The R Journal* (7:2), p. 288.

"R Consortium.," (2016). (available at https://www.r-consortium.org/; retrieved April 20, 2016).

Raymond, E. S. 2000. "The Cathedral and the Bazaar," September (available at http://www.catb.org/esr/writings/cathedral-bazaar/cathedral-bazaar/; retrieved March 1, 2016).

Riehle, D. 2010. "The Economic Case for Open Source Foundations," *Computer* (43:1), pp. 86–90 (doi: 10.1109/MC.2010.24).

R-Project. 2016. "Contributed Packages," (available at https://cran.r-project.org/web/packages/; retrieved March 1, 2016).

Spaeth, S., Krogh, G. von, and He, F. 2015. "Research Note—Perceived Firm Attributes and Intrinsic Motivation in Sponsored Open Source Software Projects," *Information Systems Research* (26:1), pp. 224–237 (doi: 10.1287/isre.2014.0539).

The R Foundation. 2016. "The R Foundation," (available at https://www.r-project.org/foundation/; retrieved March 1, 2016).

Vance, A. 2009. "Data Analysts Captivated by R's Power," *New York Times*, p. B6 (available at http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html; retrieved March 1, 2016).