Reflections on Open Source Software and Open Science Peer Production

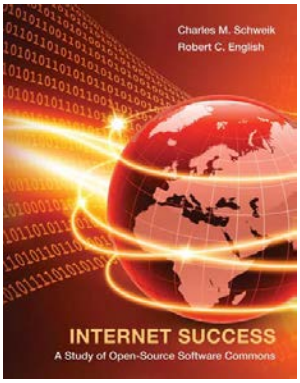
Charlie Schweik

Department of Environmental Conservation and Center for Public Policy and Administration

University of Massachusetts, Amherst

- "Peer production is the most significant organizational innovation that has emerged from Internet-mediated social practice."
- "Organizationally, it combines three core characteristics:
  - (a) decentralization of conception and execution of problems and solutions,
  - (b) harnessing diverse motivations, and
  - (c) separation of governance and management from property and contract."

Y. Benkler, 2013. Peer Production and Cooperation, In J. M. Bauer & M. Latzer (eds.), Handbook on the Economics of the Internet, Cheltenham and Northampton, Edward Elgar.

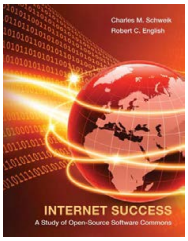


Talk Overview

1. Key Empirical Insights from my Open Source Commons-based Peer Production (CBPP) Study
2. Three cases of CBPP in open educational resources and open science/hardware
3. Puzzles and Potential Research Directions

Part I

Key Empirical Insights from my Open Source Software CBPP research

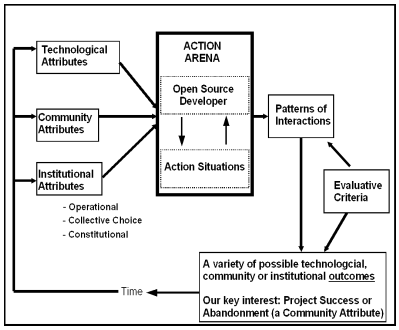


Open Source Software Empirical Study

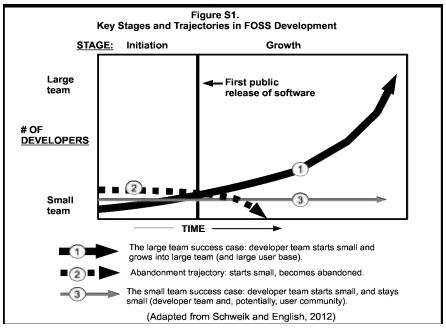
1. Dependent variable: Successful ongoing collaborations or abandoned projects
2. What factors influence project success or abandonment?

Schweik, C.M. and English, R. 2012. *Internet Success: A Study of Open Source Software Commons*. MIT Press.

Overarching Research Question:
What factors influence FOSS project success or abandonment?



The dependent variable:
Successful ongoing collaboration or abandoned



Open Source Software Empirical Study

(Schweik and English, 2012)

2009 dataset

- 174,000+ projects from Sourceforge.net
- 1400+ projects survey responses

Schweik, C.M. and English, R. 2012. *Internet Success: A Study of Open Source Software Commons*. MIT Press.

A Few Key Findings

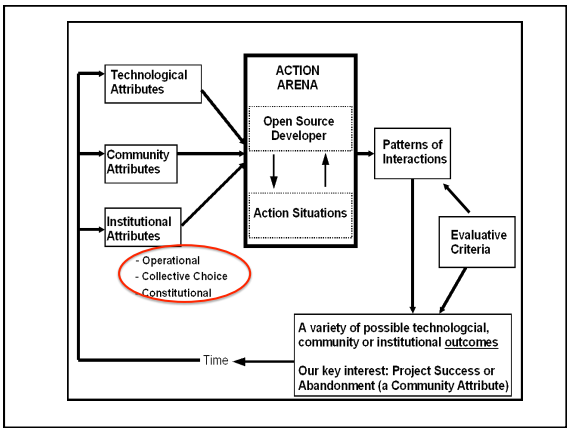
(Schweik and English, 2012)

1. A Theory of Compound Incentives

Schweik, C.M. and English, R. 2012.
Internet Success: A Study of Open Source Software Commons. MIT Press.

2. Governance/institutional designs of open source software CBPP projects and overarching nonprofit foundations

Schweik, C.M. and English, R. 2012.
Internet Success: A Study of Open Source Software Commons. MIT Press.



Governance and Team Size

Developer Count	SF Population: Class AG	SF Population: Class SG
1	37259 (48)	11765 (15)
2	8329 (11)	4420 (6)
3	3236 (4)	2380 (3)
4	1573 (2)	1515 (2)
5	921 (1)	1076 (1)
6	525 (1)	731 (1)
7	316 (0)	514 (1)
8	198 (0)	424 (0)
9	141 (0)	309 (0)
10	90 (0)	234 (0)
11–20	305 (0)	990 (1)
>20	53 (0)	457 (0)
Totals	52,946 (68)	24,815 (32)

Governance & Operational Rules:

Social norms, collaboration system

Governance & Operational Rules –

Moving toward more formalized systems

Table 5.1
Ostrom's seven general rule categories

Rule category	Definition (can apply to any or all of the three nested levels: operational, collective choice, or constitutional)
Position	Define the positions that participants hold
Boundary	Define • who is eligible to take a position (succession rules) • the process that determines which participants may enter (entry rules), such as by invitation, through some sort of competition, or compulsory • how an individual can leave a position (exit rules) Rules also may exist regarding the relationship between multiple positions, such as a mandate that no one person can hold multiple positions at the same time
Choice	Specify what participants must, must not, or may do in their positions and particular circumstances Choice rules focus on actions
Aggregation	Determine whether a decision by a single or multiple participants is needed prior to an action at a decision point in a process Aggregation rules are needed whenever choice rules provide partial control to multiple positions over the same sorts of actions Aggregation rules can be symmetrical (e.g., unanimous) or nonsymmetrical (e.g., a leader can make a decision for a group), and each of these rules must also include a nonagreement rule
Information	Specify the channels used to communicate information among participants as well as what kinds of information can be transmitted by what positions Rules also may specify required frequency of interaction or an official language
Payoff	Assign external rewards and sanctions for particular actions or outcomes (for example, some kind of payment for completion of a task)
Scope	Specify which outcomes may, must, or must not be affected or produced within a situation Scope rules focus on *outcomes* (compared to choice rules, which focus on *actions*)

Source: Adapted from Ostrom 2005, 193–210.

Table 5.3
Summary of the institutional evolution of the Debian Linux OSS commons

Institutional evolutionary phase	Defining events	Important details
Phase 1: De facto governance (1993–1997)	Autocratic leadership emerges and is challenged by participating developers	Tensions with new leader (appointed by founder) because of being perceived as taking too much control of project led to want to instead formalize roles, rights, and responsibilities
Phase 2: Designing governance (1997–1999)	Formal positions of authority developed; formal authority is limited through democratic means	Drafted constitution formalizes roles, rights, and responsibilities, and 350 developers ratify it Key provisions: • requires developers with positional power to defer to collective wishes • project leader is subject to same rules as any member • any members can propose a general resolution to counter leader's actions • technical committee established with authority over technical debates and decisions
Phase 3: Implementing governance (1999–2003)	Various conceptions of type of leader needed are debated; community leaders are elected through democratic means	Annual elections Various leadership types: • "hands-off" leadership • technical manager • visionary • organizational building Visionary and organizational types elected during these four years
Phase 4: Stabilizing governance	Shared conception of project leader type emerges	General trend appears to be organizational-building type, probably partly reflecting that a technical committee also exists (authors' addition)

Source: Adapted from O'Mahony and Ferraro 2007, 1085, table 2.

Table 5.4
Rules identified in phases 1 and 2 of the Debian Linux case

Phase 1 (de facto governance)	Phase 2 (defining governance)
Boundary rules (norms) None articulated	**Boundary rules (formal)** Constitutional level • project leaders are limited to one year • annual elections, with self-nominations allowed; three-week polling period Collective choice level • clear distinction between project leader's role and the technical committee's role
Position rules (norms) Constitutional and collective choice levels • project leader and trusted lieutenants	**Position rules (formal)** Constitutional and collective choice levels • Member/developer, project leader, and technical committee members
Choice rules (norms) Operational level • trusted lieutenants' actions (e.g., code commits) • actions permitted/ prohibited to users or developers by the OSS license	**Choice rules (formal)** Constitutional level • technical committee cannot introduce new proposals, but has the authority to resolve disputes • any member has the right to propose a general resolution that can counter a leader's actions Collective choice level • technical committee can decide any technical matter where developer jurisdictions overlap Operational level • the technical committee and project leader must make decisions that are consistent with the consensus of the opinions of the developers
Information rules None mentioned in case write-up	**Information rules** None mentioned in case write-up
Aggregation rules (norms) Operational-level • project leader responsible for dispute resolution	**Aggregation rules** Constitutional level • in order to overrule a developer, a supermajority (three-fourths) of the [technical] committee must agree • the case study suggests that there is a process for dealing with a general resolution in response to a leader's actions, but it is not described in any more detail

Table 6.1
Ostrom's seven general rule categories in OSGeo's institutional design

Ostrom's rule category	Ostrom's definition	Examples in OSGeo's institutional design
Position rules	Define the positions that participants hold	Board of directors (BOD) President and CEO Vice president Committee chair Corporate officer Member Participant
Boundary rules	Define: • who is eligible to take a position (succession rules) • the process that determines which participants may enter (entry rules), such as by invitation, through some sort of competition, or compulsory • how an individual can leave a position (exit rules) Other rules may exist regarding the relationship between multiple positions, such as a mandate that one person cannot hold multiple positions at the same time	BOD election BOD member leaving Committee chair Charter versus other members
Choice rules	Specify what participants in positions must, must not, or may do in their positions, and in particular circumstances Choice rules focus on *actions*	Bylaws for BOD Bylaws for officers Committee rules/policies Incubation process
Aggregation rules	Determine whether a decision by a single or multiple participants is needed prior to an action at a decision point in a process Aggregation rules are needed whenever choice rules provide multiple positions or partial control over the same sorts of actions Aggregation rules can be symmetrical (e.g., unanimity) or nonsymmetrical (where a leader can make a decision for a group), and each also must include a nonagreement rule	Symmetrical: Consensus in committees Nonsymmetrical: BOD creates committees

Table 6.5
Ostrom's rule categories applied to OSGeo project cases

Ostrom's rule category	Project A	Project B	Project C	Project D	Project E	Project F	Project G
Position rules	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	No formal project leader Informal lead team of three people Project steering committee member Committers	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	No formal project leader Informal lead team of four people Project management committee member Core developer (informal; often overlaps with the committee member) Developer
Boundary rules	Formal rules Community members elect to project steering committee No term limits	Formal rules	Formal rules copied from another project	Formal rules copied from another project Almost never consulted	Formal rules	Formal rules exist, but primarily depend on social norms	Formal rules, but not necessarily followed

Choice rules	Some formalized Program steering committee makes some major rules Primarily social norms Open exchange in the list Mutual expectations	Some formalized: available in the wiki Primarily social norms	Some formalized: available in the wiki Primarily social norms	Social norms	Social norms	Social norms	Formalized rules written down Program management acts, if necessary Social norms important
Aggregation rules	Informal-symmetrical: consensus in program steering committee and discussion, including developers who are not on the committee Formal voting: rarely occurs, even though formal rules stipulate it Only project steering committee members can vote	Steering committee: almost all developers are on the committee Voting: if veto, vote is used and discussion follows	Informal-symmetrical: consensus in program steering committee Formal voting: rarely occurs even though formal rules stipulate it Only project steering committee members can vote	Steering committee makes decision by consensus or voting All developers can vote as well, but their votes do not count	Informal-symmetrical: consensus in program steering committee Formal voting: rarely occurs even though formal rules stipulate it Only project steering committee members can vote	Informal-symmetrical: consensus in program steering committee and discussion, but often back channels used before the decision is reached Voting is a last resort	Program management committee votes

The tables on institutional design evolution or project comparisons...

-- they're unsatisfying.

3. Online and Face-to-face interaction and Social Capital

Schweik, C.M. and English, R. 2012.
Internet Success: A Study of Open Source Software Commons. MIT Press.

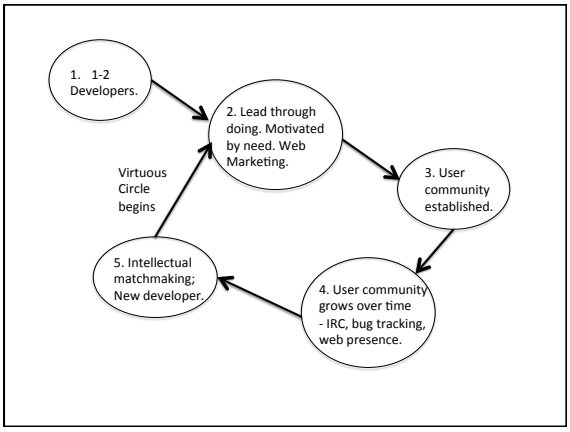
4. Power-Law Intellectual Matchmaking Hubs

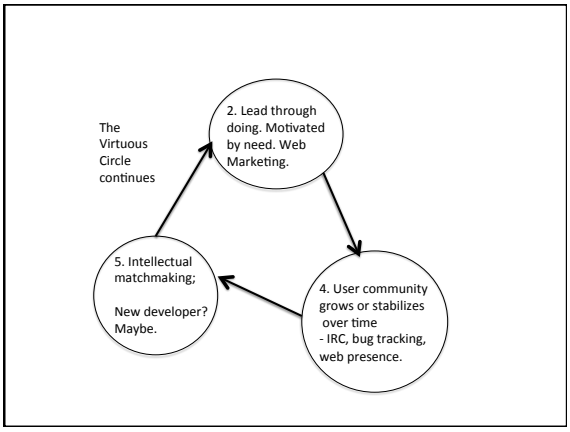
Schweik, C.M. and English, R. 2012.
Internet Success: A Study of Open Source Software Commons. MIT Press.

5. Empirical Evidence Supporting the Conventional Wisdom Open Source

-- a 'Virtuous Circle'

Schweik, C.M. and English, R. 2012.
Internet Success: A Study of Open Source Software Commons. MIT Press.





The overall story of successful collaborative open source CBPP is…

- It's really not about growing large development teams

  it's a story about

- small groups of people -- sometimes living far away from each other geographically – finding and connecting with each other, who possess similar user-centered needs, interests, passions and abilities, and all want to use or work on a particular open source software project.
- Overtime they build social capital and collaborate, and (perhaps) grow a user community.

Are these findings obvious?

-- Maybe.

But our study provides *strong empirical evidence* for these points.

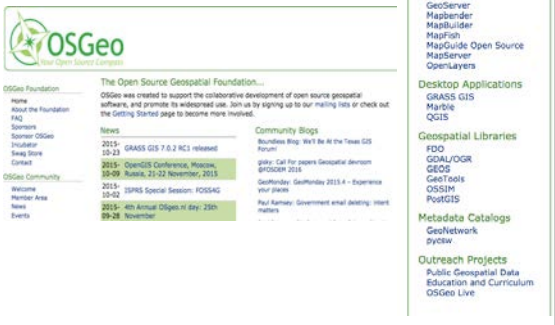
One more…

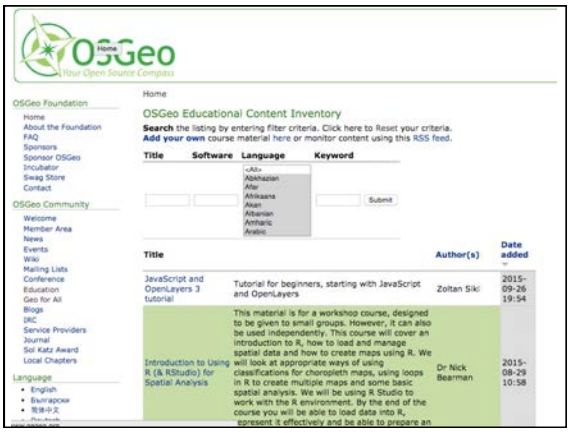
Part II

"Theory without Experience is Fantasy; Experience without Theory is Blind"

-- Vincent Ostrom
(…channeling Immanuel Kant)

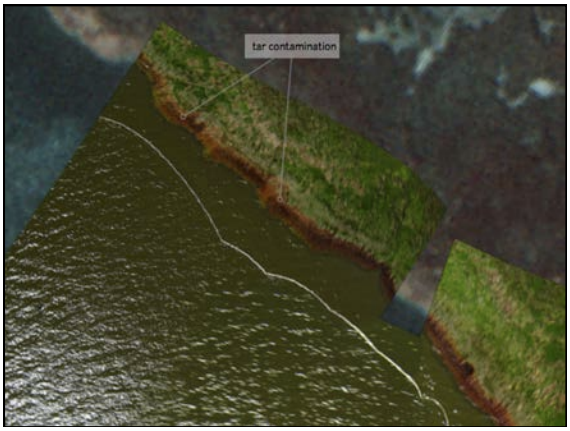
Case 1:
CBPP in Open Educational Resources

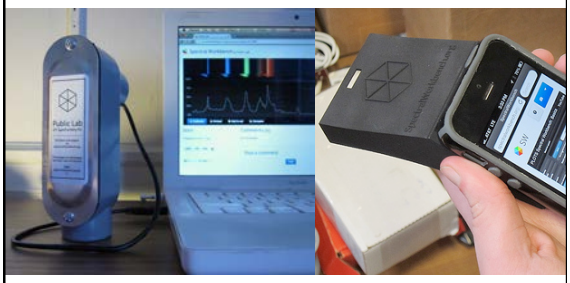




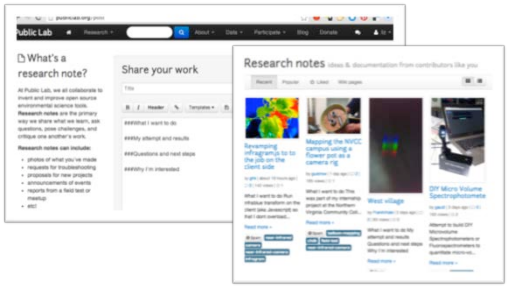
GeoForAll

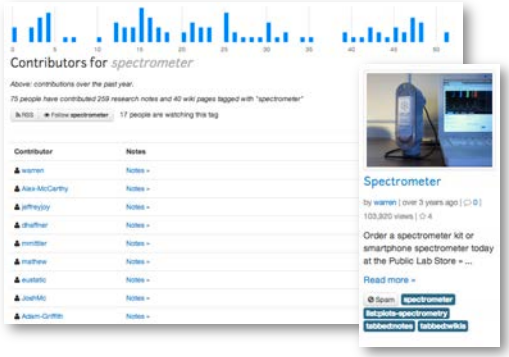
Case 2: PublicLab.org
CBPP in DIY Citizen Science/Activism





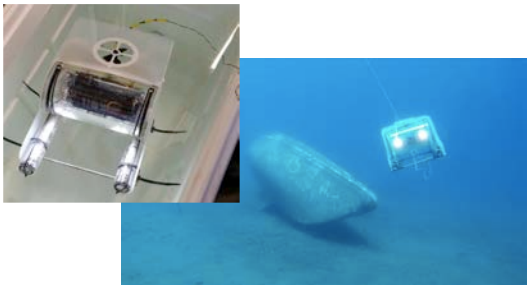
Publiclab.org's Collaboration System:
'Research Notes'

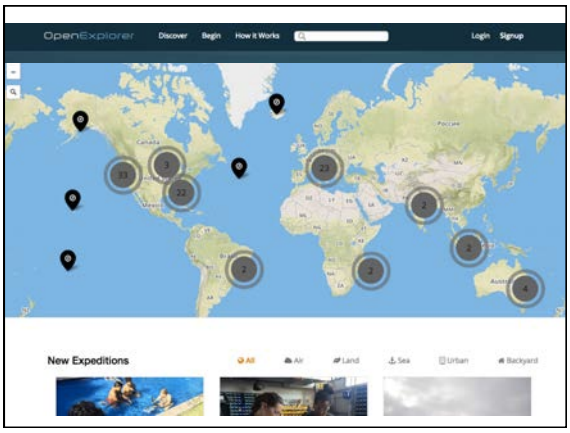




publiclab.org/**contributors**

Case 3. OpenROV
DIY Tethered Research Submarine





Reflections on Cases

- Theory of compound incentives
- Face-to-face and social capital
- Governance
- Virtuous Circle
- Power-law intellectual matchmaking
  - Water chestnut image analysis example

Part III:

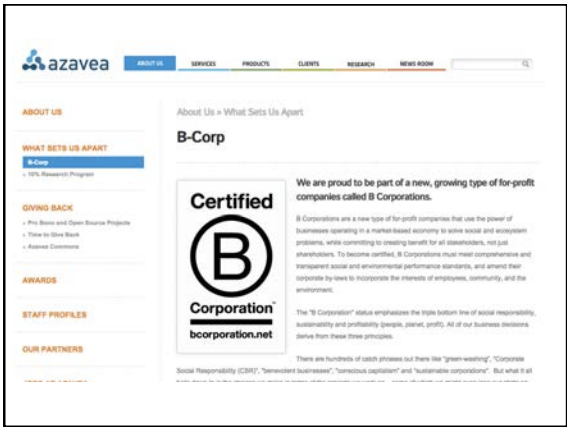
Puzzles or Possible CBPP Research Agenda Items

1. How can we study the structure and evolution of CBPP governance in a more systematic fashion that permits quantitative analysis?

Table 6.5
Ostrom's rule categories applied to OSGeo project cases

Ostrom's rule category	Project A	Project B	Project C	Project D	Project E	Project F	Project G
Position rules	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	No formal project leader Informal lead team of three people Project steering committee member Committers	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	Project leader Project steering committee member Core developer (informal; often overlaps with the committee member) Developer	No formal project leader Informal lead team of four people Project management committee member Core developer (informal; often overlaps with the committee member) Developer
Boundary rules	Formal rules Community members elect to project steering committee No term limits	Formal rules	Formal rules copied from another project	Formal rules copied from another project Almost never consulted	Formal rules	Formal rules exist, but primarily depend on social norms	Formal rules, but not necessarily followed

2. Research on the financing side of CBPP



3. How do version control systems and other technologies for collaboration and communication differ or how are they alike across different CBPP situations?

Embedded operational level rules?

4. Managing complexity in CBPP in the era of atoms and not bits

Thank you!

cschweik@pubpol.umass.edu