

Sustainable Communities and the Continuous Development of Open-Source Software

Yunwen Ye

Center for LifeLong Learning and Design

Department of Computer Science, University of Colorado at Boulder

yunwen@cs.colorado.edu; <http://www.cs.colorado.edu/~yunwen>

Research Opportunities in OSS

The wide success of OSS projects demonstrate the feasibility of creating very complicated systems of the finest quality in a distributed, collaborative, and continuously evolving manner, and provide case studies for HCI (Human-Computer Interaction), software engineering, and social science researchers as they develop, validate, and refine their theories on computer-mediated communication, coordination, cooperation and collaboration. Conversely, theories developed by researchers help to understand why OSS projects succeed, to establish theoretical and practical guidelines to sustain the continuous and stable development of OSS projects, and to create better tools to support OSS developers.

Rather than a result of intentional design, OSS currently has emerged naturally as the result of interaction of many forces brought by social ideology change (e.g., gift culture), economic development (e.g., relatively affluent software developers), and technology advancement (e.g., the Internet). Clearly, more research work is needed, from different perspectives, to understand the critical factors that make OSS successful, and how those factors interact with each other. To unravel this rather complicated socio-technical phenomenon needs a tight integration among theory building, system development, practice, and assessment. We need to apply and refine existing theories in analyzing and assessing the current practices of OSS; practicing the newly refined theories in guiding the continuous development of old OSS projects and the design of new OSS projects; and improve the theories further through the assessment of the designed OSS projects.

Theories related to OSS

The democratization of information producing and consuming. Basically, there are two models in terms of the relationship between information production and consumption (Figure 1) (Fischer, Scharff, & Ye, 2003). In Figure 1(a), a small group of elite are chosen to produce the information. While in Figure 1(b), consumers are free to contribute information. Proprietary software and the traditional publishing establishments are examples of the former, while OSS and the WWW are examples of the latter.

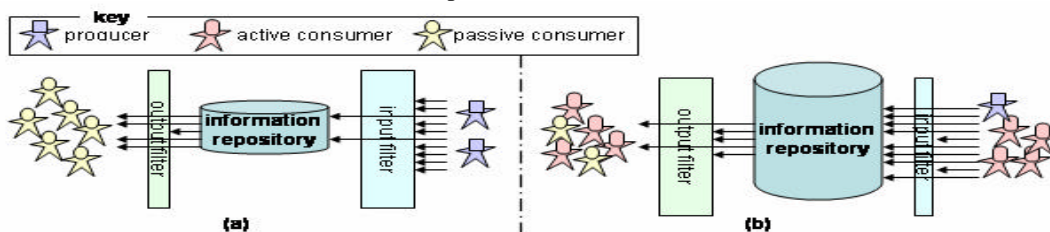


Figure 1: Two General Models of Creating and Using an Information Repository

Research issues:

- ◆ What are the requirements of technology that enables consumers to become producers? (i.e., lower the technical difficulty for contribution.)
- ◆ What are the motivations for consumers to become active?
- ◆ What are the social mechanisms to reward such active contributions? (i.e., increase the value of contribution.)
- ◆ How do you trust the information?
- ◆ How do you choose the right information for you?

Learning, and community identities in OSS development. An OSS project is unlikely to sustain a long-term success unless there is an associated community that provides the platform for developers, users, and user-turned-developers to collaborate with each other. Existing community theories provide a conceptual framework for understanding OSS. Grounded in the community learning theory called Legitimate Peripheral Participation (Lave & Wenger, 1991), we argue that learning is one the major motivations for OSS developers (Ye & Kishida, 2003).

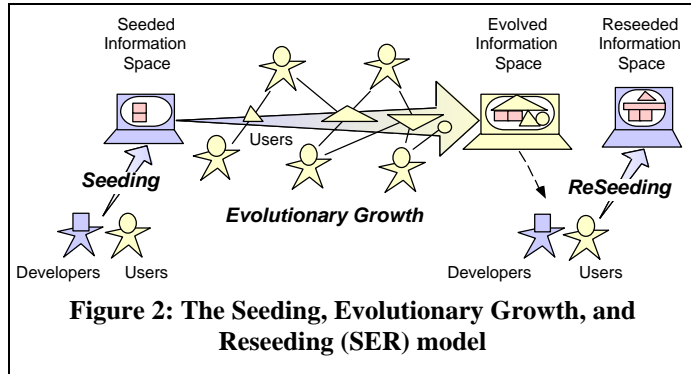
Continuous development of OSS requires the design of the community that provides learning opportunities to newcomers.

Research issues:

- ◆ A quantitative validation of this theory.
- ◆ How learning is taking place in current successful OSS communities?
- ◆ What are the effective strategies to promote learning?
- ◆ Impacts on the education of software engineering?

The Seeding, Evolutionary development, and Reseeding model.

SER model (Figure 2) (Fischer et al., 2001) that we have developed provides a conceptual framework to understand how systems like OSS projects evolve at the hands of active consumers. A seed, created by developers, is the initial state of a system that is intended to evolve. The evolutionary growth phase is one of unplanned evolution as the seed is used by the users to do their work. During this phase, the seed plays two roles: it provides resources for work and it accumulates the products of work. Reseeding is a deliberate effort to organize, formalize, and generalize information created and captured during the evolutionary growth phase.



Research issues:

- ◆ What are the requirements for a successful seed in OSS? (e.g. scratching enough itches? Providing enough learning opportunities?)
- ◆ How to manage the evolutionary growth?
- ◆ When to reseed and how to recognize that there is a need for reseeding?

Some research issues that we are working on

Understanding the Co-evolution of OSS systems and OSS communities

Our previous research (Ye & Kishida, 2003) has found that there are eight different roles in OSS communities. The role transformation and reconstitution that evolves the OSS community leads to the evolution of the OSS system, and the evolution of OSS system caused by the contribution of a community member leads the role change of the member (Figure 3).

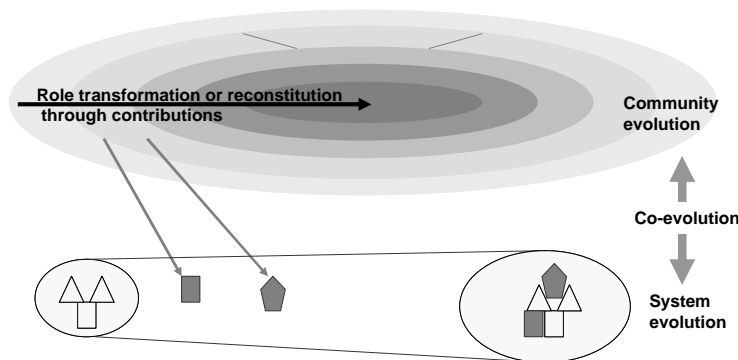


Figure 3: The Co-Evolution of F/OSS Systems and F/OSS Communities

Creating A taxonomy for OSS projects

We have identified that at least three different types of F/OSS projects exist: *Exploration-Oriented*, *Utility-Oriented*, and *Service-Oriented* (Table 1) (Nakakoji, Yamamoto, Nishinaka, Kishida, & Ye, 2002). We do not mean that the three types cover all the F/OSS projects. Our attempt at defining three types of F/OSS is to create a general understanding that although all of these systems are called free or open-source software, differences do exist in primary objective, control style, system evolution, community structure and evolution.

Table 1: Three Types of OSS Projects

| Type | Objective | Control style | System evolution pattern | Community structure | Major problems | Examples |
|----------------------|-----------------------------------|-----------------------------------|---|---|---------------------------------------|-----------------------------------|
| Exploration-Oriented | Sharing innovations and knowledge | Cathedral-like central control | Single branch Feedback from the community | Project Leader Many Readers | Subject to split | GNU systems Jun |
| Utility-Oriented | Satisfying an individual need | Bazaar-like decentralized control | Multiple versions coexist Tournament style | Many Peripheral Developers Peer support to Passive Users | Difficult to choose the right program | Linux system excluding the kernel |
| Service-Oriented | Providing stable services | Council-like central control | Single branch Patches merged through control | Core Members instead of a Project Leader Many Passive Users that develop systems for end-users | Less innovation | PostgreSQL Apache |

Understanding the dynamic changes of OSS projects

The type of an OSS project may evolve, as the primary objective of, and other factors affecting, the OSS project change over the time. Our working hypothesis is that Exploration-Oriented OSS and Utility-Oriented OSS are good for the initiation of an OSS project, and Service-Oriented OSS is suitable for more mature OSS projects (Figure 4).

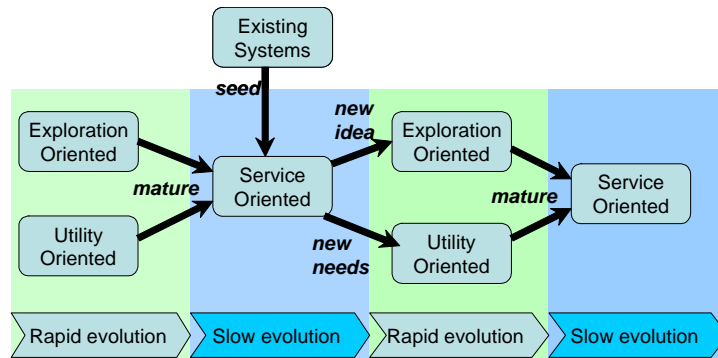


Figure 4: The Evolution Pattern of OSS Projects

References: (available from <http://www.cs.colorado.edu/~yunwen>; or <http://www.cs.colorado.edu/~gerhard>)

Fischer, G., Grudin, J., McCall, R., Ostwald, J., Redmiles, D., Reeves, B., & Shipman, F. (2001). Seeding, Evolutionary Growth and Reseeding: The Incremental Development of Collaborative Design Environments. In G. Olson & T. Malone & J. Smith (Eds.), *Coordination Theory and Collaboration Technology* (pp. 447-472). Mahwah, NJ: Lawrence Erlbaum.

Fischer, G., Scharff, E., & Ye, Y. (2003). Fostering Social Creativity by Increasing Social Capital. In M. Huysman & V. Wulf (Eds.), *Social Capital* (pp. (in press)).

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK: Cambridge University Press.

Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K., & Ye, Y. (2002, May 19-20). Evolution Patterns of Open-Source Software Systems and Communities. *Proceedings of International Workshop on Principles of Software Evolution (IWPSE 2002)*, Orlando, FL, 76-85.

Ye, Y., & Kishida, K. (2003). Toward an Understanding of the Motivation of Open Source Software Developers. *Proceedings of 2003 International Conference on Software Engineering (ICSE'03)*, Portland, OR, 419-429.