OSS 2008
Milan, September 7-10 2008

Reflection on Knowledge Sharing in F/OSS Projects

Sulayman K. Sowe & Ioannis Stamelos

sksowe@csd.auth.gr

Dept. of Informatics, Aristotle University, Thessaloniki

International conference on Open Source Systems,

September 7th -11th , 2008, Milan.

In this Reflection.....

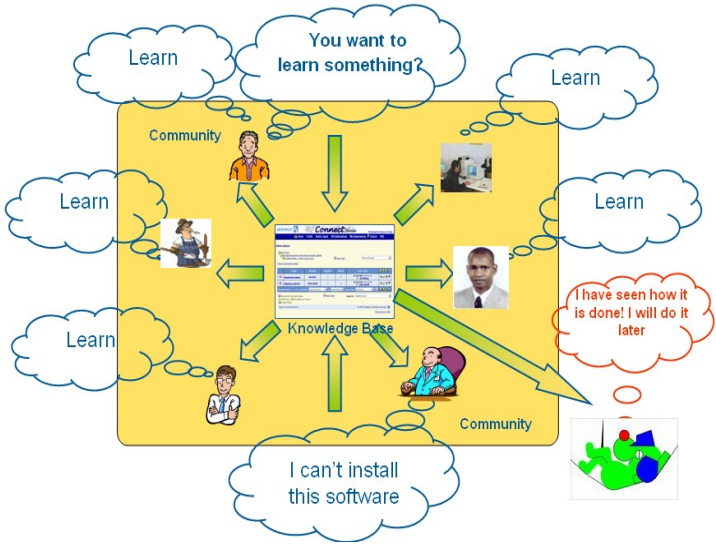
- **Theme 1: Knowledge sharing in distributed software development**
  - Knowledge sharing: personal view
  - Importance of knowledge sharing for F/OSS projects
  - Some knowledge sharing problems
- **Theme 2: Knowledge Sharing enablers in F/OSS projects**
  - Availability of software code/knowledge
  - Community dynamics
- **Theme 3: Paradigm Shift in knowledge management practices**
  - Knowledge sharing in closed-source environments
  - Knowledge sharing in open-source projects
- **Theme 4: An empirical study**
  - Findings & Implications
- **Theme 5: Reflection on knowledge sharing practices**

T1: Knowledge sharing in distributed software development

- Knowledge sharing:
  - A synergistic process where project participants establish knowledge links (**k**) by "talking to each other", and in the process get more than they put in:

    $\mathbf{k}_{AB}$ = 1 *if there is knowledge sharing between actors* **A** *and* **B**, *and* 0 *if otherwise*.

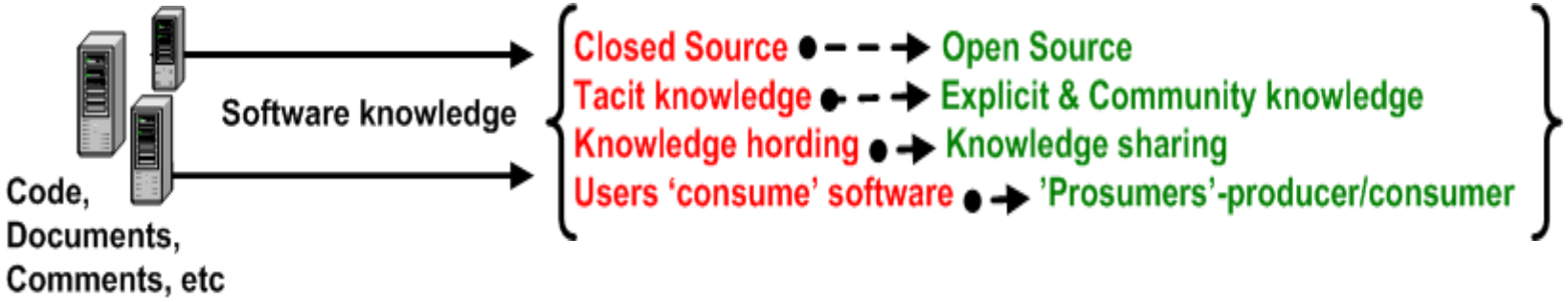
- Importance of knowledge sharing for F/OSS projects:
  - Produce goods (software) and services (support).
  - Develop, enhance and refine project strategies.
  - Enable individual and project learning.
  - Build trust and confidence amongst participants.



- Some knowledge sharing problems:
  - How can projects leverage and transform the tacit knowledge of community members into explicit usable knowledge?
  - How to coordinate individuals or even infrastructures which are often located at large distances from each other?
  - How to achieve f2f knowledge sharing? Sprints, conferences, etc.?
  - How to provide easy to use tools for light-weight knowledge sharing?
  - How to accommodate viewpoints of a diverse group?

T2: Knowledge sharing enablers in F/OSS projects

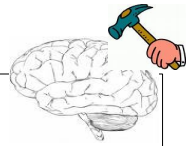
- Tools (forums, mailing lists, SVN, etc.) to facilitate communication and coordination.



- But
  - Is there knowledge sharing?
  - Who are the people involved?
  - How much knowledge sharing are developers and users doing?
  - What is being talked about?
  - How will those 'excluded' from talking feel?

T3: Paradigm shift in KM practices

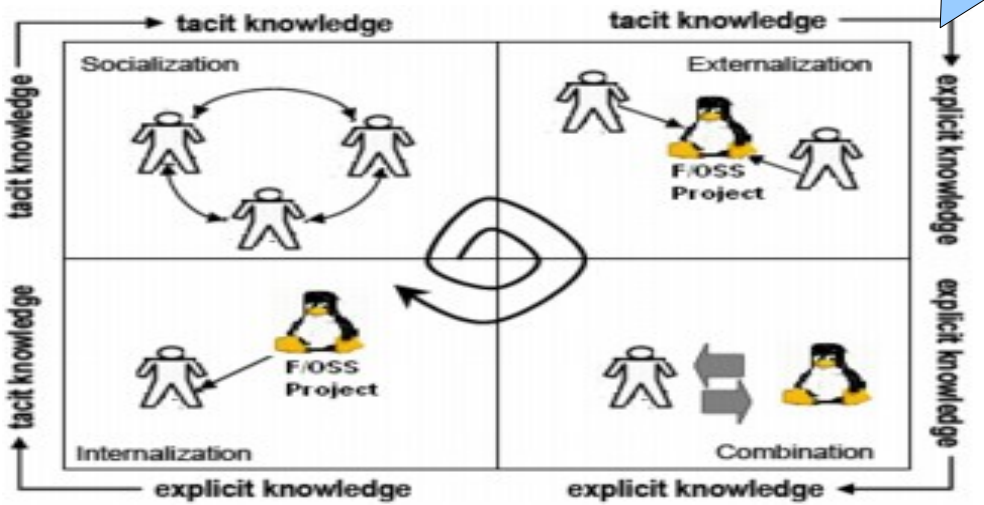
- The goal: lower barriers imposed by organizational structures (virtual or physical) so that the generation, acquisition, transfer, and sharing of knowledge can be facilitated.



- Closed-source systems:
  - Source of knowledge (code) is guarded secret.
  - Knowledge can be traded and sold.
  - New staff learn from documented practices of old staff. But how much have they documented?
  - Strict schedules and deadlines may not be conducive for effective knowledge sharing.
  - Employees leave and “take” their knowledge along with them.

- Interactive F/OSS systems:
  - Source of knowledge (code) is open to all for scrutiny
  - Share knowledge; selling difficult because someone might provide the same knowledge for free.
  - Newbies learn from experts, archives and documented practices and know how
  - Learn, use, and share whenever and wherever you want.
  - From Tacit to Explicit knowledge: volunteers externalize their knowledge in forums, doc., CVS/SVN for future participants to lean from.

Tacit-Explicit in F/OSS projects



T4: An Empirical Investigation....1

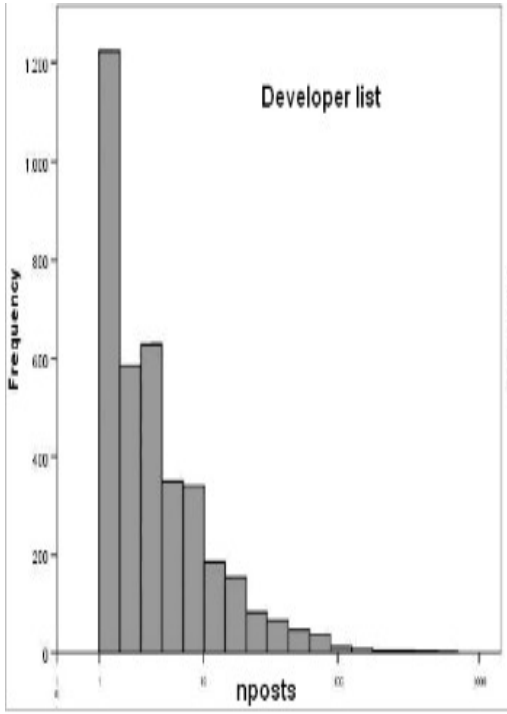
- Research venues; studying knowledge sharing in F/OSS projects:
  - Source (control) code management systems (CVS/SVN)
    - Count developers who made commits to the same module in a project
  - Projects documentation
    - Count individuals who collaborated in writing/editing a document, website, etc.
  - **Mailing lists**
    - Total number of emails posted to a list (***nposts***)
    - Total number of replies made to questions posted to the lists (***nreplies***)
      - Our data set:- Debian developer and users mailing lists (01/01/00-31/12/05)
        - Developer list: N = 3735 participants; *nposts*= 29685; *nreplies*= 128933.
        - User lists: N = 5970 participants; *nposts*= 193276; *nreplies*= 165380.

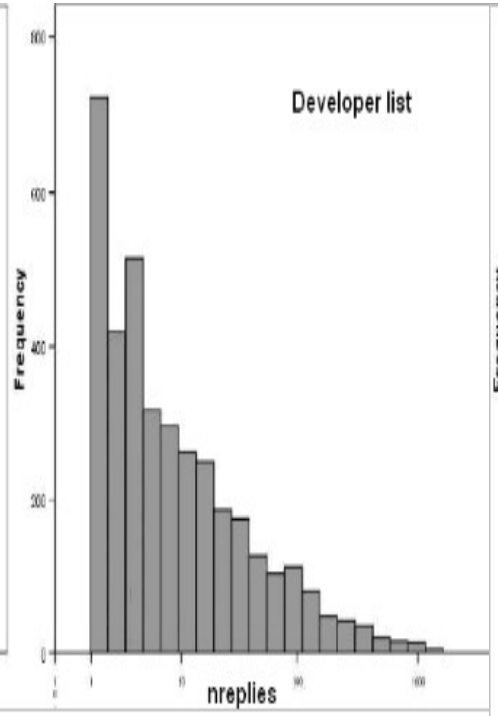
		Developer	User
nposts	Mean	7.95	**32.37**
	Median	3.00	7.00
	Std. Dev.	21.302	121.753
	Maximum	523	4106
nreplies	Mean	**34.52**	27.70
	Median	6.00	5.00
	Std. Dev.	105.57	122.04
	Maximum	1517	4168

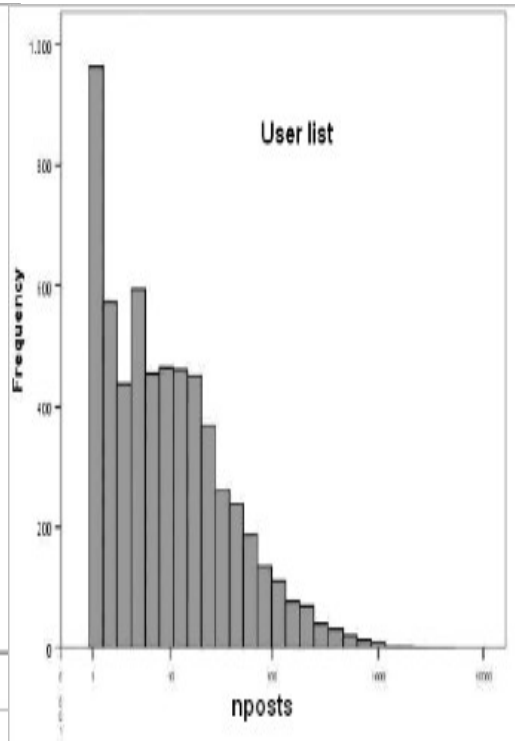
Observation:
- Posting and replying activities skewed.
- Posting and replying activities different for different lists
- Mean (posts/person) and median of *nposts* are smaller for the Developer list, while the same measures have larger values for *nreplies*.

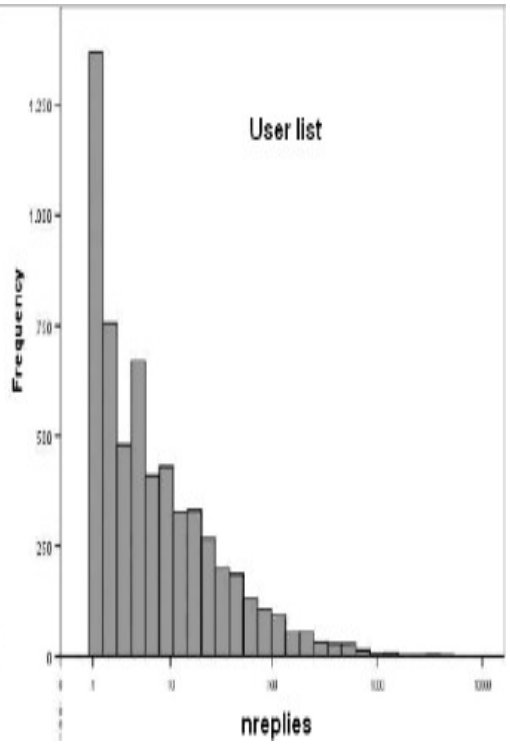
Theme 4: An Empirical Investigation....2

- Developer list participants: small number of posts and a large number of replies.
- User list participants: small posts and small replies.









- Relationship: posting and replying activities are highly related, corr higher for User list.

			Developer List		User List	
Test	Variable		nposts	nreplies	nposts	nreplies
Kandella's *taub*	nposts	Corr. Coef.	1000	,475*	1000	,550*
	nreplies	Corr. Coef.	,475*	1000	,550*	1000
Spearmans *rho*	nposts	Corr. Coef.	1000	,608*	1000	,699*
	nreplies	Corr. Coef.	,608*	1000	**,699***	1000

Theme 5: Reflection on knowledge sharing practices

- In the Developer list participants contributed more replies:
  - messages contain sufficient information, given the highest priority, receive the attention of almost all participants.
- User list participants posted more than they replied to questions asked in the lists:
  - Too much 'noise', novice and some uninteresting messages, important requests may be lost or not spotted early
- Star posters are also star repliers:
  - Knows project and software,
  - longtime project associates.
    - But this may have Implications for information overload, and
    - valuable coding time may be lost
- Crests and troughs (Wave) effects
  - When posts increase, replies increase correspondingly

•What needs to be done:

- More qualitative studies to study software developers and users online and offline
- Are developers coding as much as they are 'talking' in lists? See WoPDaSD workshop paper

Thanks a lot for your attention

Questions & Comments