Towards A review of the empirical FLOSS literature

Kevin Crowston, James Howison, Kangning Wei, U. Yeliz Esereyli, Qing Li

Syracuse University
Method for review paper

- Review of current research
  - What is known, where are the gaps?
- Looked for papers in Web of Science, ABI/Inform, journal special issues, AOM and AIS conferences, IntOSS (IFIP) conferences, ICSE workshops, opensource.mit.edu
- Restricted review to empirical papers on FLOSS development or use
- Found more than 500 papers in 1st pass
- Need to do a further pass to capture literature of past 18 months!
Literature Review Summary

✓ 586 articles reviewed
  ✓ 295 Both Empirical and Relevant
    ✓ Irrelevant excluded studies which simply used open source software for analysis or proof of concept
✓ Tagged in hierarchical categories
  ✓ Level of Analysis
  ✓ Method
  ✓ Projects studied
  ✓ Data Type
  ✓ Constructs
  ✓ Discipline
✓ Tried tagging for theory, but almost one per paper!
Analysis technique (Clustering)
Demo
Relevant and Empirical Articles Per Year

Clearly Truncated in early 2006!

More articles to collect

1997 is a Roy Fielding working paper
Papers by Method used

- 150 In-depth study of small number of projects
  - 102 Case-study
  - 21 Interviews (non-case study)
  - 21 Participant observation
  - 11 Ethnographies
  - 6 Discourse analysis

- 70 Large scale sample measurement
  - 37 Surveys
  - 32 Other quantitative (eg correlation models)

- 13 Review Article
- 11 Not Specified/Unclear
- 6 Action Research/Design Science
- 4 Experiment
Other nuggets

✓ Top Projects Studied
  ✓ 59 Linux
  ✓ 32 Apache
  ✓ 22 Mozilla
  ✓ 13 Gnome
  ✓ 10 Debian
  ✓ Very long tail

✓ Very few longitudinal studies (~10)
  ✓ Many others compounded data over lifetime but didn’t study time-series or change

✓ About the same number studied for
  ✓ Motivations, Coordination and Decision Making
Constructs

- Schema reorganization is ongoing, show current version but unfortunately without exact counts
- Inputs (~50%)
- Process (~30%)
- Outputs (~30%)
- Some studies included all three (therefore > 100%)
Inputs

✓ Individuals
  ✓ Characteristics
  ✓ Motivations
  ✓ Contributions (inc Time spent and Roles)

✓ Companies
  ✓ Characteristics, Motivations and Contributions

✓ Teams
  ✓ Project Characteristics (License etc)
  ✓ Membership (Div of labor, distribution of effort)
  ✓ Technology use
  ✓ Interaction Structures (eg Onion Model)
Contribution: So much by so few….

The graph shows the proportion of posts contributed by each developer. The x-axis represents the developers, and the y-axis represents the proportion of posts. The developers are listed in descending order of their contribution, with the most active at the top. The graph illustrates the concentration of contributions among a small number of developers, with a long tail of less active contributors.
Source: Sourceforge Weekly Download Stats, Week 1 2005.
With file sharing removed, the emphasis is tools that help technical tasks.

Source: Sourceforge Weekly Download Stats, Week 1 2005.
Developer numbers

67% never more than 1 developer, only 1.9% have had >10 developers
Processes (I)

✓ General Team Processes
  ✓ Socialization
  ✓ Governance (Group Decisions) and Leadership
  ✓ Coordination
  ✓ Team Maintenance
  ✓ Knowledge Management
Process (II)

✓ Software Development Practices
  ✓ Requirements
  ✓ Planning and Design
  ✓ Coding
  ✓ Releases
  ✓ Maintenance
    ✓ Change cycle (bugs and features)
    ✓ User support
  ✓ Project Management
Public and Private Cycles in Development
Outputs

- Performance measures
  - System Quality (highest number)
  - Use (e.g., downloads)
  - User Satisfaction (few, using Freshmeat)
  - Impacts (Individual and Organizational)
    - Eg Learning or Revenue/Costs
  - Processes (e.g., bug-fixing speed)
  - Project member satisfaction

- Antecedents of performance

- Evolution
  - Of the software artifact
  - Of the team and its practices
Observations

✓ More work done than expected, more done since then!
✓ Less bias towards motivations than expected
✓ Substantial and cumulating body of work on software quality and other output measures
✓ Need to develop a shared taxonomy of organizational types
✓ Little longitudinal work, but it is very revealing, especially for taxonomy (Different paths to success and failure)
To Do

✓ Add recent literature
  ✓ May do this during review, since one is always behind the curve

✓ Considering normalizing tags
  ✓ So if study covers 5 topics in minimal depth, counts for less in an area than a paper in depth on single topic

✓ Considering tagging for contribution
  ✓ Difficult to assess quality, will probably just do this in the text of the paper.

✓ Tags by year
  ✓ Show the movement of research into different topics

✓ Finish text and finalize venue for submission
  ✓ Considering MISQ Review and ACM Computing Surveys