Evolution in Free and Open Source Software: A Study of Multiple Repositories

Karl Beecher, University of Lincoln, UK

Freie Universität Berlin
Germany
25 September 2009

Outline

- Brief Introduction to FOSS
- Observations
- Questions
- Approach & Results
- Interpretation & Conclusions
- Further Work

Brief Introduction to FOSS

- “Free” or “Open Source” Software
  - The freedom to run the program, for any purpose
  - The freedom to change the program to make it do what you wish
  - The freedom to redistribute copies
  - The freedom to release your improvements to the public

Brief Introduction to FOSS II

- Anything more to be said about FOSS?
  - Raymond – observations and “tips” on FOSS development [Ray01]
  - Godfrey & Tu – growth of software [God00]
  - Mockus et al – structure of distributed development teams [Moc02]
  - German – software “archaeology”
  - Capiluppi – FOSS evolution [Cap04]

Observations

Preliminary research that led to the formation of the ”problem”.

Observations II

- Plenty of non-empirical work.
- Many works examining one or a small number of deliberately chosen FOSS projects.
- FOSS “collections” rarely studied.
- Most works using endogenous characteristics of software. Exogenous characteristics rarely studied.

Questions

The research ”problem” phrased as some unresolved questions requiring investigation.

Questions II

- Are FOSS projects influenced by the environment in which they are developed?

  “communities/repositories” [Sca04]

- Can an evidence-based framework be developed to describe the effects?

Approach and Results

The method developed to answer these research questions.

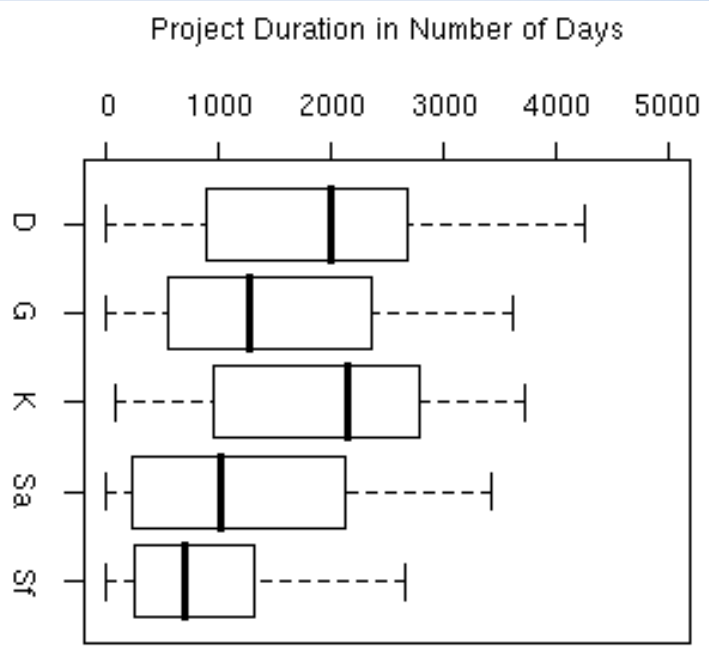
The outcome of the investigations.

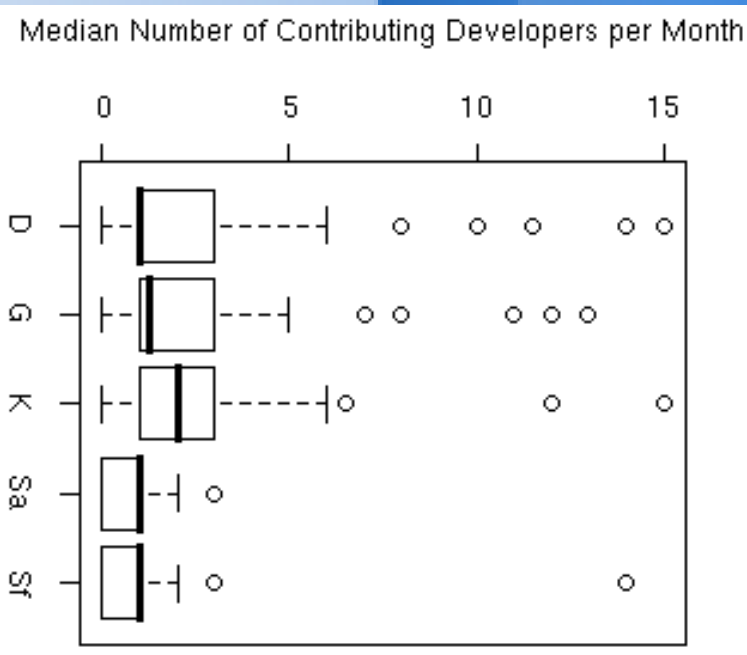
Preliminary Study

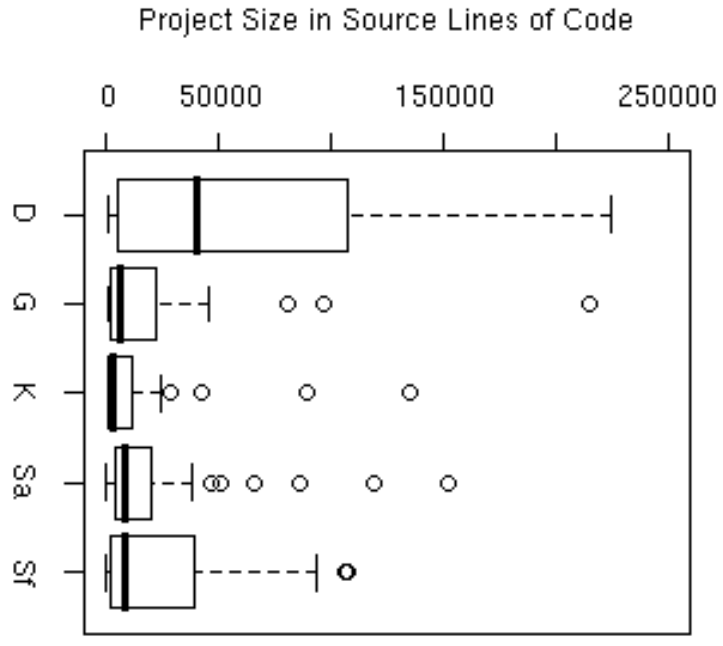
- A selection of five FOSS repositories.
  - Differing levels of “prestige”, i.e. barriers to entry, organizational control, quality requirements, wideness of distribution.
  - A randomized sample of 50 projects per repository downloaded (historical).
  - Evolutionary metrics of projects:
    - Average monthly number of developers and commits
    - Size
    - Age

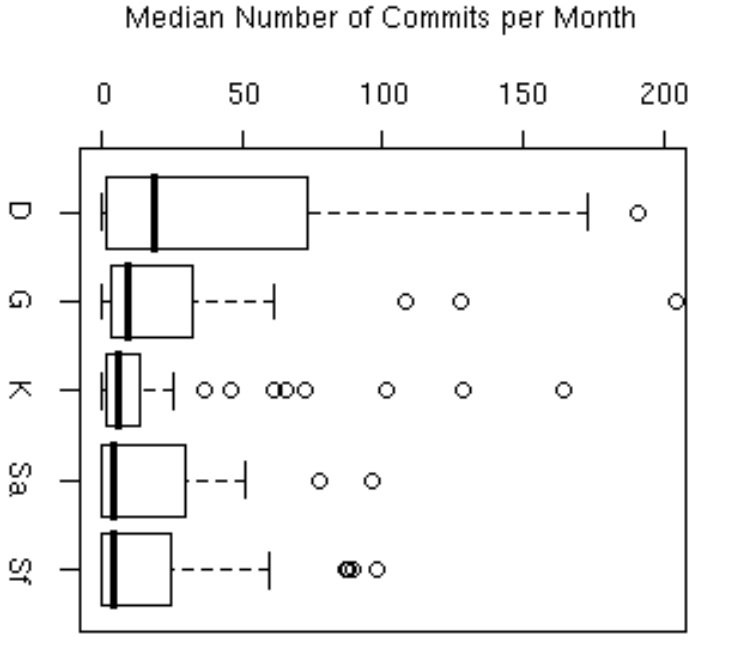
Preliminary Study - Repositories

- Debian
- GNOME
- KDE
- Savannah
- SourceForge


Project Duration in Number of Days
0
1000
2000
3000
4000
5000
D
G
K
Sa
Sf


Median Number of Contributing Developers per Month
0
5
10
15
D
G
K
Sa
Sf


Project Size in Source Lines of Code
0
50000
150000
250000
D
G
K
Sa
Sf


Median Number of Commits per Month
0
50
100
150
200
D
G
K
Sa
Sf

Preliminary Study - Results

- The repositories appeared to group for many of the measures:
  - Group 1: Debian, GNOME and KDE.
  - Group 2: Savannah and SourceForge.
- Group 1 had more “successful” evolutionary attributes than group 2.

Taking the Study Further

- [Bee09]
- Questions:
  - Were the differences significant, or were they the results of statistical noise?
  - Is the grouping valid?
- Approach for first investigation:
  - Each repository sample couple compared using statistical significance testing.
    - (Wilcoxon unpaired rank-sum test with Bonferroni correction.)

Taking the Study Further II

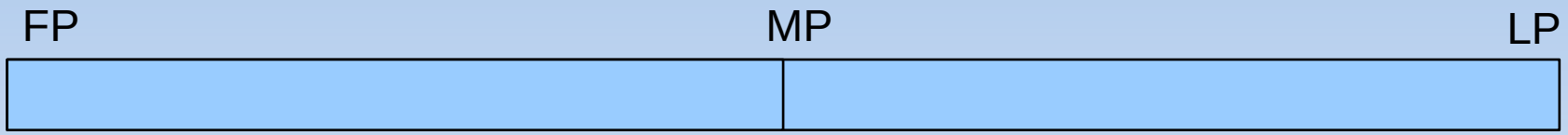
- Results of the statistical significance test revealed:
  - Differences between Group 1 and Group 2 were significant in a majority of cases.
- Furthermore:
  - KDE and GNOME samples demonstrated notable similarity, as did Savannah and SourceForge samples.
  - Within Group 1, Debian sample attributes were more “successful” than others, a little more often than not.

Next Investigation – Evolutionary Study

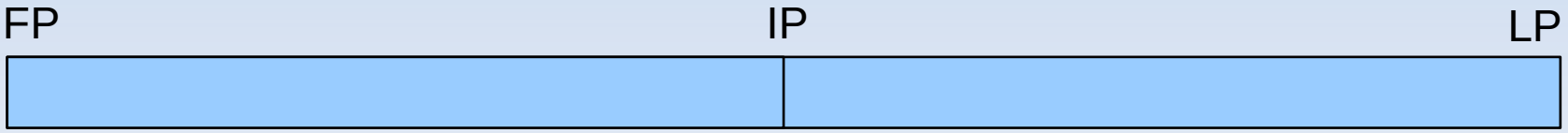
- [Bee07, Cap09]
- In each repository, where is the evolutionary effort going?
  - Lehman tells us that software complexity must be controlled for it to evolve satisfactorily.
  - Do more evolved projects receive greater amounts of such complexity control work?
- Furthermore:
  - Debian is a *distribution*.
  - Some projects in the other repositories transit into Debian.
  - Can we observe a “transition effect”?

Evolutionary Study II

- Sub-sample of projects from each of the 5 repositories, 3 historical snapshots, equally spaced



- Sub-sample of Debian projects only, 3 historical snapshots, based on insertion into Debian



  - Rate of commits and developers
  - Rate of control work done for McCabe complexity and functional instability

Evolutionary Study – Results

	FP → MP	MP → LP
Debian	7.3%	5.5%
GNOME	6.1%	6.6%
KDE	7.7%	8.2%
Savannah	2.7%	2.8%
SourceForge	1.5%	0.8%

Average percentage of functions receiving work done to reduce instability in the period described.

Evolutionary Study – Results

- For projects from Debian sample:
  - Rate of developers: FP→IP ≤ IP→LP for 82% of projects. Remaining projects stayed the same.
  - Rate of commits: FP→IP ≤ IP→LP for 55% of projects. Remaining project stayed the same or decreased.
  - Product measures (McCabe & functional instability):
    - 2/3 of projects received greater rates of complexity control work after IP (both measures)
    - 1/6 received greater rates in only one measure
    - The remainder saw no increase.

Interpretation and Conclusions

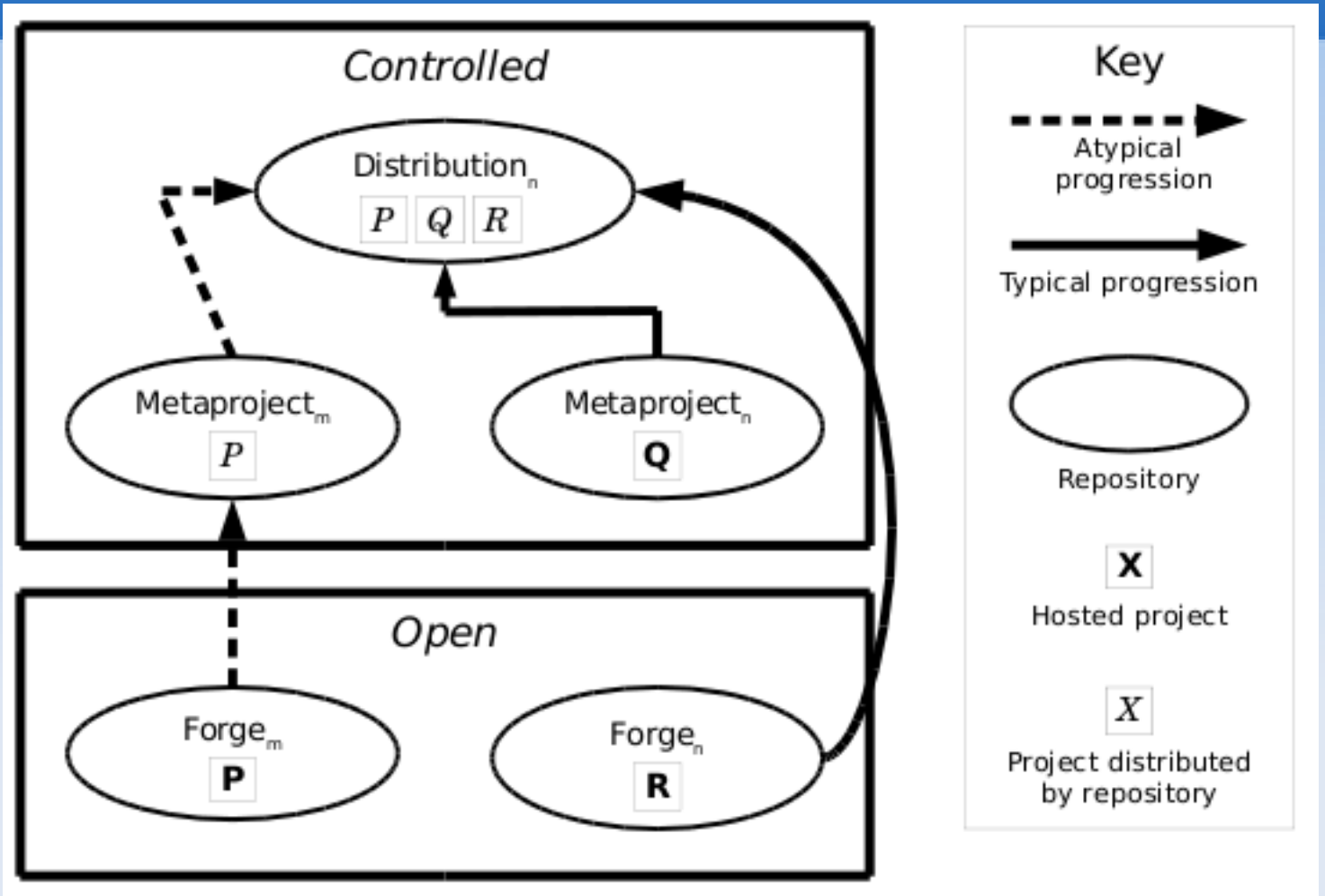
What does the work conclude?

How can the results be interpreted? In particular, can they be modeled/formalized?

Interpretations and Conclusions II

- Chosen repositories are divided according to measured attributes of projects.
- Evidence supports the view that they are an environmental factor in project evolution.
- Repositories are similar according to both the measured attributes and “prestige”.

Evolutionary Framework



Further Work

- Enriching data sets.
- Widen repository collection.
- More formal deliverables.
- Evaluation and diagnosis for evolvability:
  - Automatic/semi-automatic analysis.
  - Identification of poorly evolving areas.
  - Diagnosis for improving evolution.

Further Resources

- http://cross.lincoln.ac.uk
  - Home of the CROSS team (inc. Cornelia Boldyreff, Andrea Capiluppi, Paul Adams and me).
- http://floss-ori.org
  - New research initiative for providing open access to FOSS research, a central point for knowledge exchange, and a crossover between academia and industry.
- http://www.floss.lincoln.ac.uk
  - Personal blog for (mostly) FOSS work.

References

- [Bee07] K Beecher, C Boldyreff, A Capiluppi, and S Rank. Evolutionary success of open source software: An investigation into exogenous drivers. Electronic Communications of the EASST, 8, http://www.easst.org/eceasst, 2008. Reprinted. Originally published in the Proceedings of the ERCIM Symposium on Software Evolution, co-located with the ICSM 2007 Conference, 2007.
- [Bee09] K Beecher, A Capiluppi, and C Boldyreff. Identifying exogenous drivers and evolutionary stages in floss projects. Journal of Systems and Software, 82:739–750, May 2009.
- [Cap07] A Capiluppi and M Michlmayr. From the cathedral to the bazaar: An empirical study of the lifecycle of volunteer community projects. In Proceedings of the 3rd International Conference on Open Source Systems, Limerick, Ireland, June 2007.
- [Cap09] A Capiluppi, K Beecher. Structural complexity and decay in FLOSS systems: an inter-repository study. Proceedings of the 13th European Conference on Software Maintenance and Reengineering, Kaiserslautern, Germany, 2009.
- [God00] M Godfrey and Q Tu. Evolution in open source software: A case study. In Proceedings of 16th IEEE Int. Conf. on Software Maintenance, San Jose, California, USA, October 2000.
- [Moc02] A Mockus, R Fielding, and J Herbsleb. Two case studies of open source software development: Apache and mozilla. ACM Transactions of Software Engineering and Methodology, pages 309–364, July 2002.
- [Ray01] ERaymond. The cathedral and the bazaar. In The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O' Reilly, 2001.
- [Sca04] W Scacchi. Free/open source software development practices in the computer game community. IEEE Software, pages 59–66, January 2004.

Thank you for your attention.