



# CS 7460 Collaborative Computing: Open Source Software

*Munmun De Choudhury*

[munmund@gatech.edu](mailto:munmund@gatech.edu)

Week 4 | January 29, 2015

# Changes in Class Schedule

- Check the weeks you have signed up to lead discussion – probably is misalignment due to the changes.

“Release Early, Release Often” –  
Why do you think it works?

Crowston et al. found that FLOSS projects do not have a formal planning or software requirements analysis phase. Why are they still successful?

Traditional Software  
development suffers from  
Brook's Law. Why is that not a  
problem in OSS?

Raymond argues that Brook's law indicates how the high cost of communication in groups hurts software development. Why is communication not a problem in OSS?

FLOSS suffers from all the limitations of distributed work. What reasons do the authors cite that still makes it successful?

One can test, debug and improve  
in bazaar style. But *cannot* code  
group up in bazaar style. Why?  
What would be the problem?

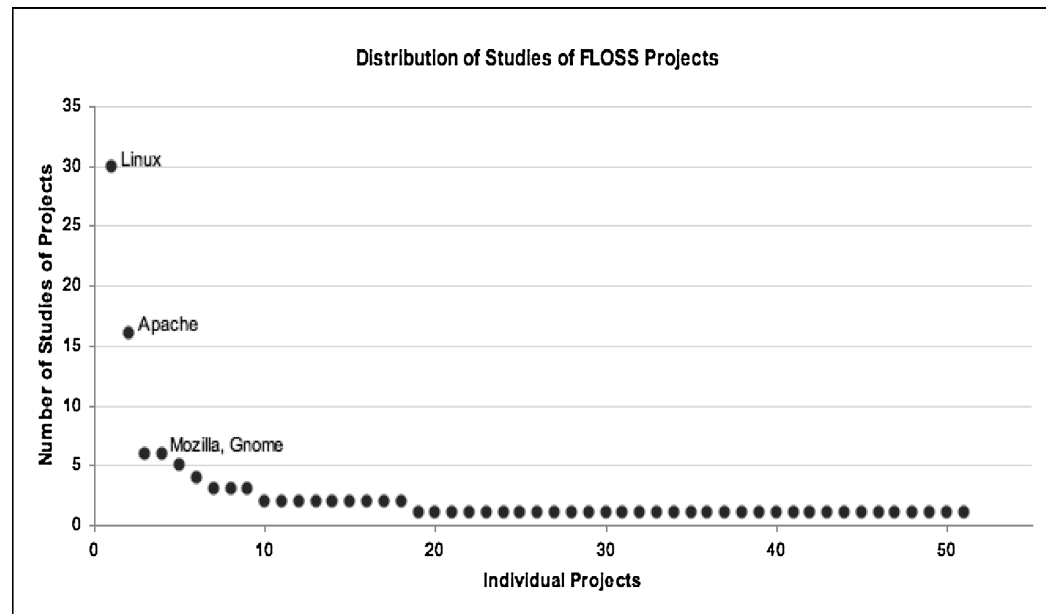
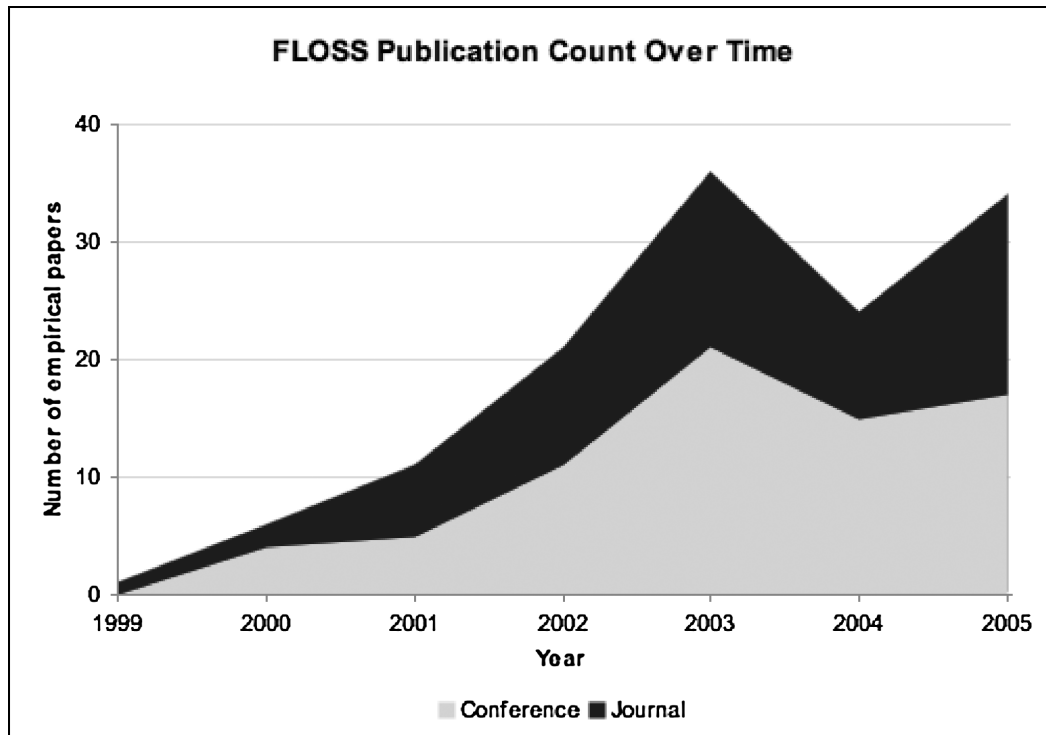


Raymond argues that “The open-source community's internal market in reputation exerts subtle pressure on people not to launch development efforts they're not competent to follow through”. This is a huge challenge of open source collaboration. What are its strengths and limitations?

There is no monetary compensation in FLOSS. How do they still manage to attract and retain contributors?

Raymond uses the “free market” analogy from economics to describe the success of Linux. Ego-boost and reputation turn to be major motivators for participation and success of OSS projects. Why aren't these elements for the cathedral style of software development?

Cleverness and design originality should be restrained for successful OSS projects. Sounds counter-intuitive, why?



Crowston et al. found that results on license choice and FLOSS effectiveness is mixed. Why would this be the case? [Restrictive – more productivity and membership; non-restrictive – more popularity]