Hierarchy/Core in OSS

1. **crowston˙core˙2006**
    - Provides examples of Äcademic case studies of FLOSS Projects" that have hierarchical structures
    - Ät the center of the onion are the core developers, who contribute most of the code and oversee the design and evolution of the project. In the next ring out are the co-developers who submit patches (e.g., bug fixes), which are reviewed and checked in by core developers."
2. **crowston˙hierarchy˙2006**
    - FLOSS projects are not uniformly centralized/hierarchical
    - Projects are highly connected
    - Lots of one way communication in projects
    - Projects become more decentralized as project size grows (expect there to be more write rank contributors as project size increases)
3. **wessel˙github˙2023**
    - Categorizes actions across four categories: utilities, continuous integration, code quality, and deployment
        - Look into other tables from https://link.springer.com/article/10.1007/s10664-023-10369-w/tables/1
    - increase in % of merged PRs (but then decreases over time), decrease in % of unmerged PRs (continues to decreases over time)
    - decrease in comments in merged PRs (but increases over time), increase in comments in unmerged PRs (that also decreases over time)
    - faster close to merged PRs (but gets slower over time), faster close to unmerged PRs
    - more commits during review (but decreases over time) for merged PRs, more commits during review for unmerged PRs
    - Does project-level analysis, does not consider heterogeneity between programmer rank
4. **rashid˙exploring˙2017**

- OSS project contributors, particularly new ones, encounter increased knowledge acquiition costs when turnover occurs

5. **avelino˙abandonment˙2019**

    - Examine how knowledge acquisition costs change when projects are "abandoned", by examining truck factor detachment
    - con for my work is tat you can't measure hierarchical changes in totally defunct projects

6. **hata˙characteristics˙2015**

    - How to Contribute document is very important for projects

7. **rigby˙quantifying˙2016**

    - Truck factor might exaggerate the damage caused by developers leaving, using git blame (to see who changed last line of code in a file) might be better.

8. **mcdonald˙performance˙2013** - OSS contributors care more about project visibility, contributor count than code quality as a metric of success.