Latent Social Structure in Open Source Projects

Philipp Brüschweiler

ETH Zürich

March 16, 2010

Paper by Christian Bird, David Pattison, Raissa D’Souza, Vladimir Filkov and Premkumar Devanbu
Research Question

- Number of interactions grows quadratically with team size
- Divide and conquer

Open Source Software (OSS) projects not formally organized (Latent social structure?)
Research Question

- Number of interactions grows quadratically with team size
  - Divide and conquer
- Open Source Software (OSS) projects not formally organized
Research Question

- Number of interactions grows quadratically with team size
  - Divide and conquer
- Open Source Software (OSS) projects not formally organized
- Latent social structure?
  - Not explicit, but observable
Studied Projects

Ant
Apache
Python
Perl
PostgreSQL
Project Selection Criteria

- Well-known and stable projects
Project Selection Criteria

- Well-known and stable projects
- Complex codebases with several subsystems
Project Selection Criteria

- Well-known and stable projects
- Complex codebases with several subsystems
- Different governance structures
  - Foundation (Apache and Ant)
Project Selection Criteria

- Well-known and stable projects
- Complex codebases with several subsystems
- Different governance structures
  - Foundation (Apache and Ant)
  - Community (PostgreSQL)
Project Selection Criteria

- Well-known and stable projects
- Complex codebases with several subsystems
- Different governance structures
  - Foundation (Apache and Ant)
  - Community (PostgreSQL)
  - Monarchist (Python and Perl)
Data Mining

- Build social network of mailing list participants
Build social network of mailing list participants
Download and parse mailing list archives
  - Reconstruct threads of conversation
  - Answers to emails ↔ create link between authors
Build social network of mailing list participants

Download and parse mailing list archives
  - Reconstruct threads of conversation
  - Answers to emails ↔ create link between authors

Extract code information
  - Author and time of commit
  - File names, file contents
Data Mining

- Build social network of mailing list participants
- Download and parse mailing list archives
  - Reconstruct threads of conversation
  - Answers to emails ↔ create link between authors
- Extract code information
  - Author and time of commit
  - File names, file contents
- Time intervals of 3 months
Network partitioned into groups
Finding Community Structure – Modularity

- Network partitioned into groups

- **Modularity**: \( \sum_i \frac{\text{inside}_i}{n} - \left( \frac{\text{all}_i}{n} \right)^2 \)
  - \(\text{inside}_i\): number of connections inside group i
  - \(\text{all}_i\): number of all connections to or from group i (including \(\text{inside}_i\))
  - \(n\): total number of connections

Intuition: ratio of connections inside groups vs. between groups
Finding Community Structure – Modularity

- Network partitioned into groups
- **Modularity**: $\sum_i \frac{\text{inside}_i}{n} - \left( \frac{\text{all}_i}{n} \right)^2$
  - $\text{inside}_i$: number of connections inside group $i$
  - $\text{all}_i$: number of all connections to or from group $i$ (including $\text{inside}_i$)
  - $n$: total number of connections
- Intuition: ratio of connections inside groups vs. between groups
Values between 0 and 1
- 0: not modular
- 1: disconnected complete graphs
Finding Community Structure – Modularity

- Values between 0 and 1
  - 0: not modular
  - 1: disconnected complete graphs
- Modularity of known modular networks ranges from 0.3 to 0.7
Finding Community Structure – Modularity

- Values between 0 and 1
  - 0: not modular
  - 1: disconnected complete graphs

- Modularity of known modular networks ranges from 0.3 to 0.7

- Find partition of the network that yields highest modularity
Finding Community Structure – Modularity

- Values between 0 and 1
  - 0: not modular
  - 1: disconnected complete graphs

- Modularity of known modular networks ranges from 0.3 to 0.7

- Find partition of the network that yields highest modularity
- NP-complete, approximation used
Example Network

This network has modularity 0.39
Hypothesis 1

Mailing list participants spontaneously form subcommunities and the modularity values of these subcommunities will be significant.
Strong Community Structure

- Very significant when compared to random network
- Hypothesis 1 confirmed
Product and Process Messages

- **Product** messages
  - About code
Product and Process Messages

- Product messages
  - About code
- Process messages
  - Everything else, e.g., high-level architecture discussions
Product and Process Messages

- **Product** messages
  - About code
- **Process** messages
  - Everything else, e.g., high-level architecture discussions
- Automatic classification by scanning for names of files, functions, classes, ...
Hypothesis 2

Modularity values of networks constructed from only product messages will be higher than when only process messages or all messages are used.
Hypothesis Confirmed

- Hypothesis 2 confirmed
- Successful projects focus into subcommunities for product-related work
Hypothesis 3

Pairs of developers within the same subcommunity will have more files in common than pairs of developers from different subcommunities.
Defining Collaboration

- Compare number of files worked on by developers in
  - the same subcommunity
  - different subcommunities
Hypothesis Confirmed

- Hypothesis 3 confirmed
- Social interaction linked with programming collaboration
Subcommunities are Focused

Hypothesis 4

Subcommunities focus their attention to small parts of the system, so the average directory distance of files worked on by a subcommunity will be small.
Directory Distance

- Directory distance is the tree distance in the directory tree
Directory Distance

- **Directory distance** is the tree distance in the directory tree
- Find average directory distance of files that were worked on by a subcommunity
Directory Distance

- **Directory distance** is the tree distance in the directory tree.
- Find average directory distance of files that were worked on by a subcommunity.
- Compare to random samples of developers.
Hypothesis Not Confirmed

- No significantly lower directory distance
- Hypothesis 4 not confirmed
Hypothesis Not Confirmed

- No significantly lower directory distance
- Hypothesis 4 not confirmed
- Possible explanations:
  - Hypothesis incorrect
Hypothesis Not Confirmed

- No significantly lower directory distance
- Hypothesis 4 not confirmed
- Possible explanations:
  - Hypothesis incorrect
  - Directory distance no good measure for task focus
Conclusion

- OSS projects have strong social structures
Conclusion

- OSS projects have strong social structures
- Code discussion more modular than general discussion
OSS projects have strong social structures
Code discussion more modular than general discussion
Social interaction is linked with programming collaboration