Focus Shifting Patterns

of OSS Developers and Their Congruence with Call Graph

Qi Xuan, Aaron Okano, Premkumar Devanbu, Vladimir Filkov

University of California, Davis, CA

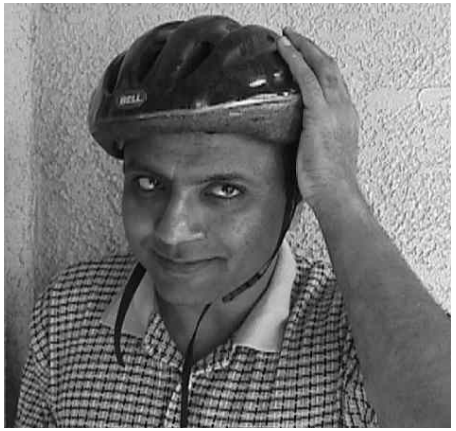
Zhejiang University of Technology, Hangzhou, China

Email: *qxuan@ucdavis.edu*

Contributors

Qi Xuan
@CS.UCDavis, @ZJUT

Aaron Okano
@CS.UCDavis



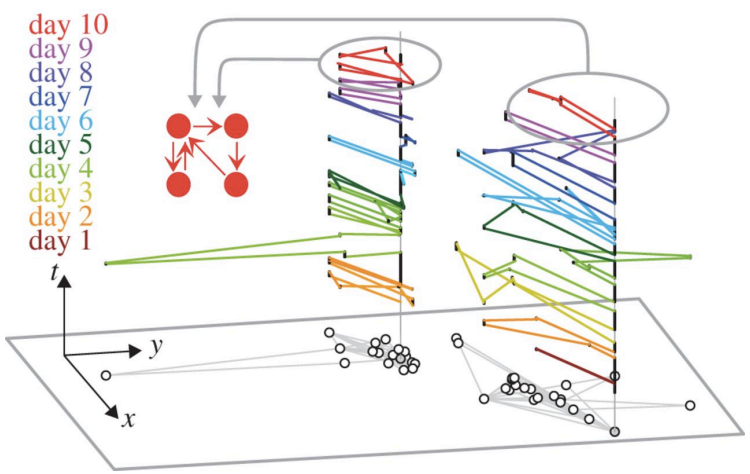
Premkumar Devanbu
@CS.UCDavis

Vladimir Filkov
@CS.UCDavis

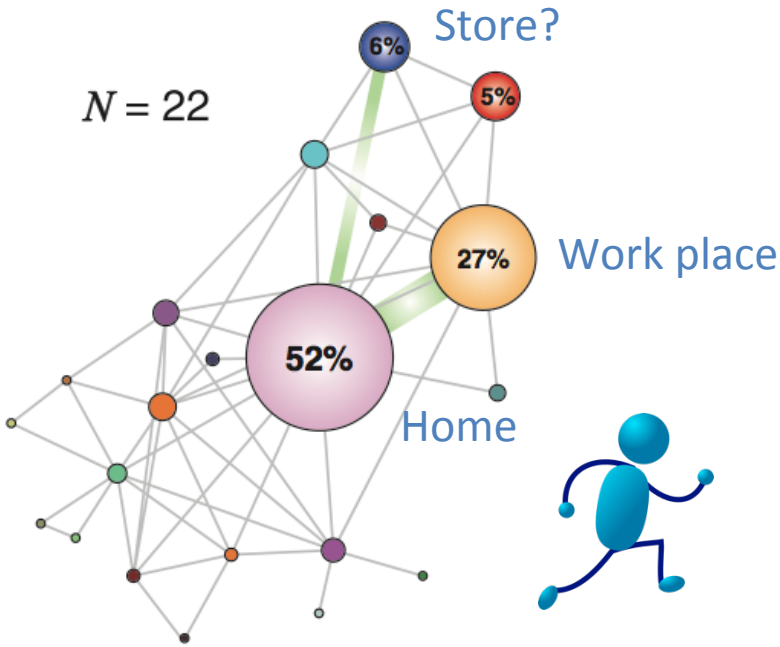
Human Mobility

Mobility Patterns

People move a lot on average.



Schneider et al. Royal Society Pub., 2013

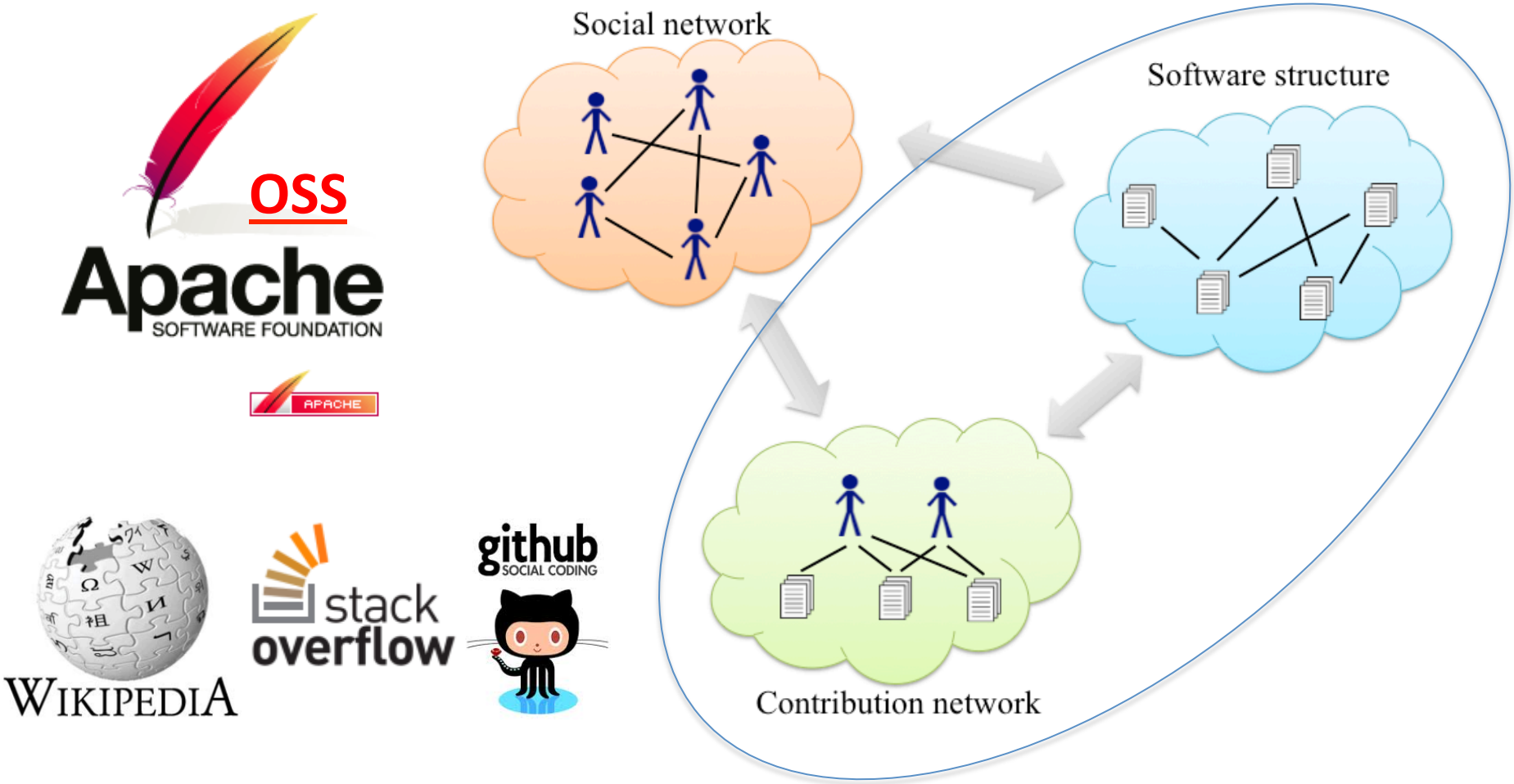


Song et al. Science, 2010

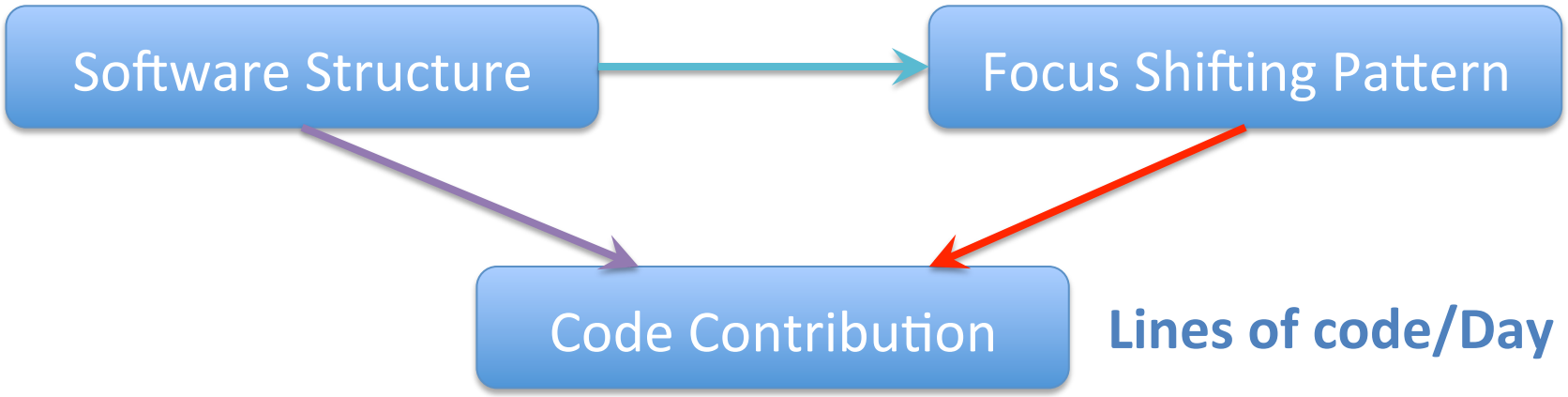
But most people move in low entropy patterns.

They are highly predictable!

Data Set


OSS
Apache
SOFTWARE FOUNDATION
APACHE
WIKIPEDIA
stack
overflow
github
SOCIAL CODING
Social network
Software structure
Contribution network

Research Questions



RQ1: Do developers shift focus through file dependency links?

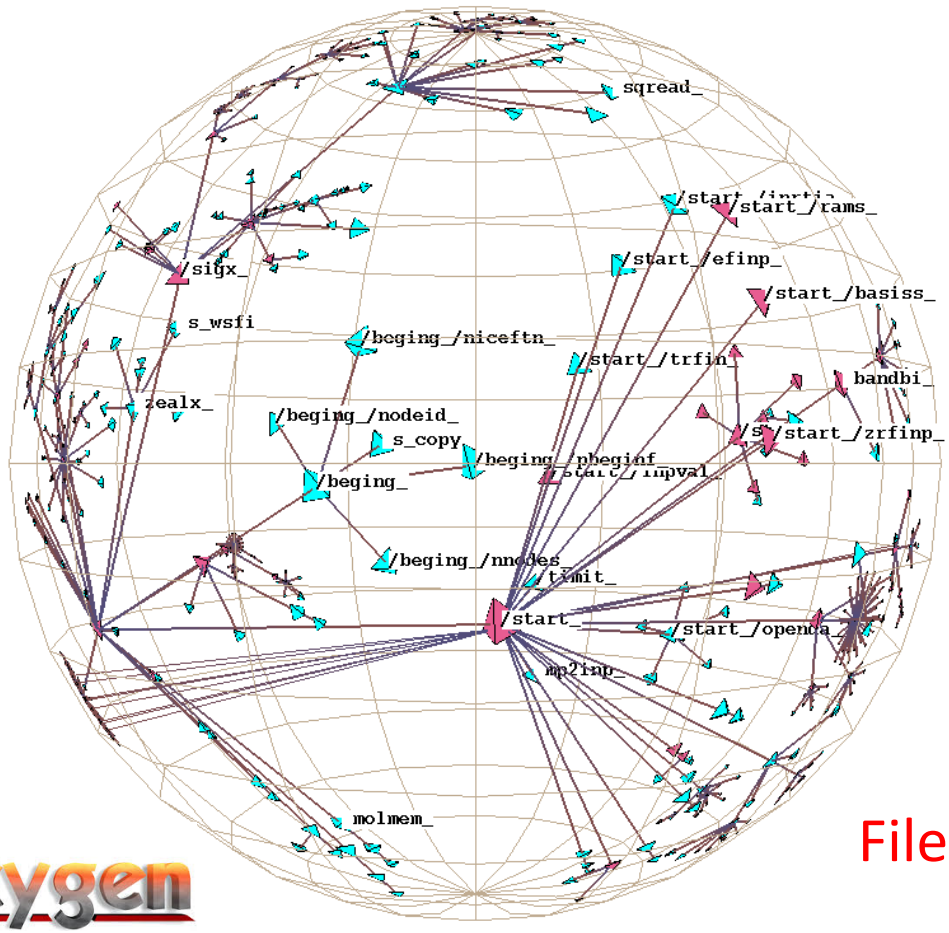
RQ2: Do developers in the same project have more similar FSPs?

RQ3: Is FSP correlated with code contribution?

RQ4: Does software structure influence code contribution?

RQ1: Do developers shift focus through file dependency links?

File Dependency Network

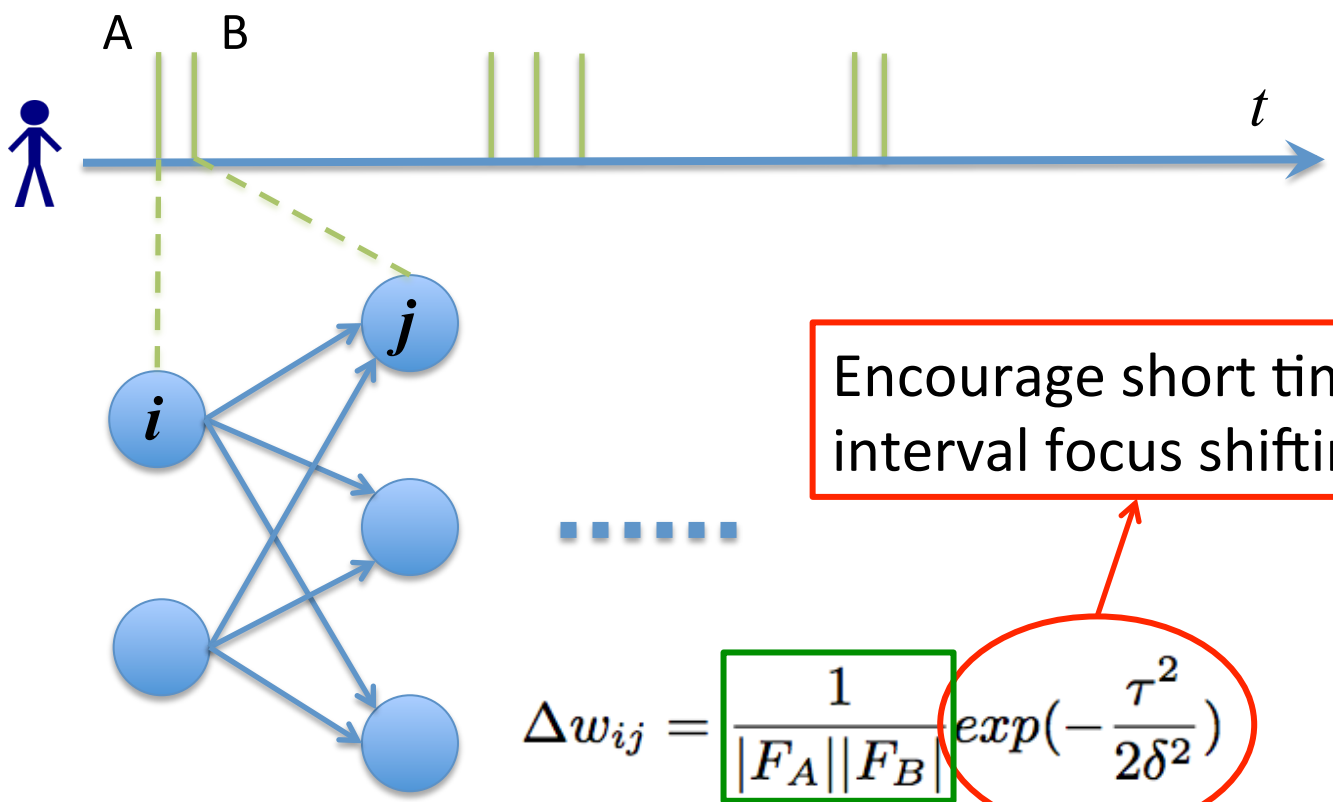


Call graph

File: A group of functions

File Dependency Network (FDN)

Focus Shifting Network



Encourage short time interval focus shifting

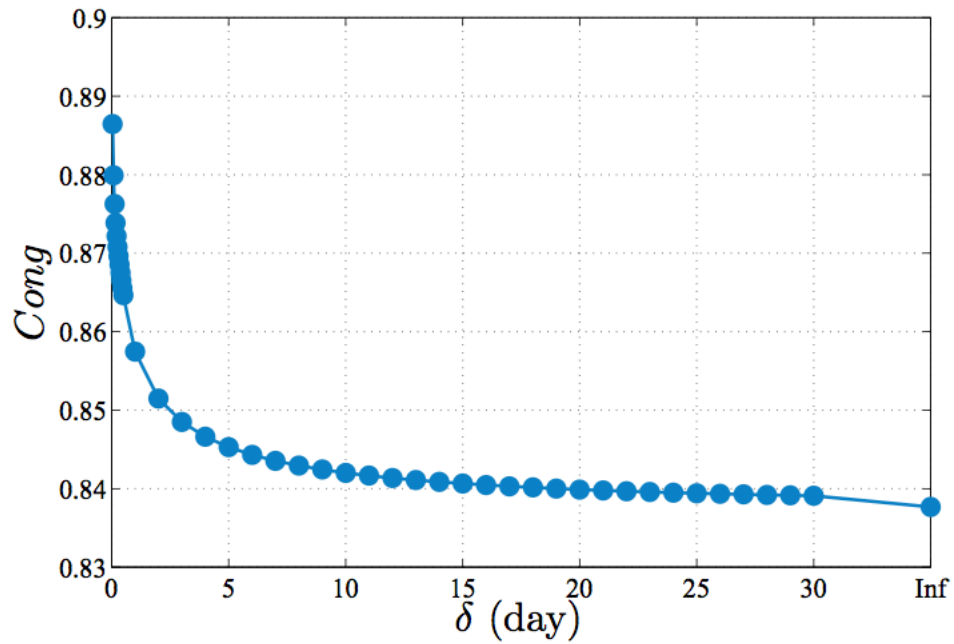
$$\Delta w_{ij} = \frac{1}{|F_A||F_B|} exp(-\frac{\tau^2}{2\delta^2})$$

W_β: weights of the FSN links that are in the FDN

W_α: weights of the FSN links that are **not** in the FDN

$$Cong = \frac{\langle W_\beta \rangle}{\langle W_\alpha \rangle + \langle W_\beta \rangle}$$

$$W_\beta > 4W_\alpha$$



Yes! ***Developers are more likely to shift focus along FDN links, especially when the interval between the commits is shorter.***

Similar method to find socio-technical congruence

Cataldo et al. ToSE, 2013

Illustration #1

https://issues.apache.org/jira/browse/DERBY-5236

K. A. Hatlen (in Derby) added code to file DRDAConnThread.java to provide a warning when a string is truncated (Sep 9, 06:47:30).

In particular, he added ……, and also added an argument to *writeFdocaVal* and modified each call to that function.

However, he discovered later that the fix introduced a bug when communicating with older clients and disabled the warning in that case. At his next commit (Sep 10, 07:00:38), he added conditionals to the function *writeLDString* in file DDMWriter.java.

The Git log commit message indicates that the modified *writeFdocaVal* calls to *writeLDString* introduced the bug.

RQ2: Do developers in the same project have more similar FSPs?

Measure FSP

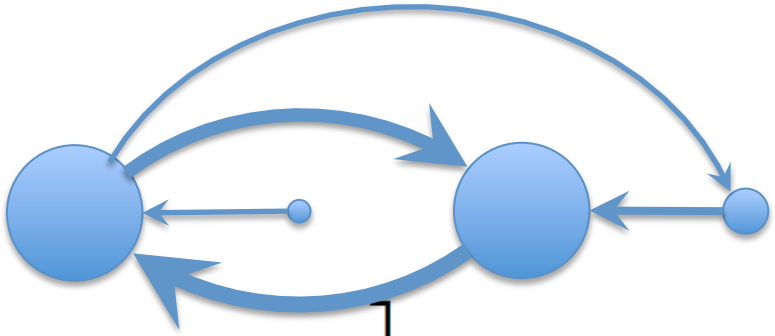
Random Entropy:

$$E_R = \log_2 N$$

Uncorrelated Entropy:

$$E_U = -\sum_{i=1}^{N} p_i \log_2 p_i$$

Markov Entropy:



$$E_M = -\sum_{i=1}^{N} \left[p_i \sum_{j \in \pi_i} p(j|i) \log_2 p(j|i) \right]$$

Similar FSPs within the Same Project

Yes! Developers' FSPs, characterized by any of the entropy, are more similar within the same project, while they are relatively different across different projects.

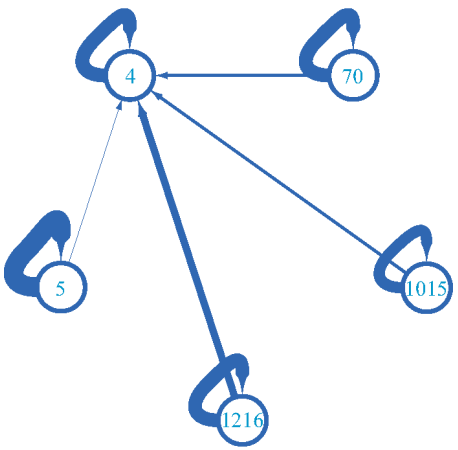
Challenge: all the three types of entropy are related to network size, so such results may just mean that the developers in the same projects tend to touch a similar number of files.

Orthogonal Decomposition:

Distributional Complexity: $\mathbf{P} = \mathbf{E_U} - \frac{\langle \mathbf{E_U}, \mathbf{E_R} \rangle}{\|\mathbf{E_R}\|^2} \mathbf{E_R}$

Structural Complexity: $\mathbf{Q} = \mathbf{E_M} - \frac{\langle \mathbf{E_M}, \mathbf{E_R} \rangle}{\|\mathbf{E_R}\|^2} \mathbf{E_R} - \frac{\langle \mathbf{E_M}, \mathbf{P} \rangle}{\|\mathbf{P}\|^2} \mathbf{P}$

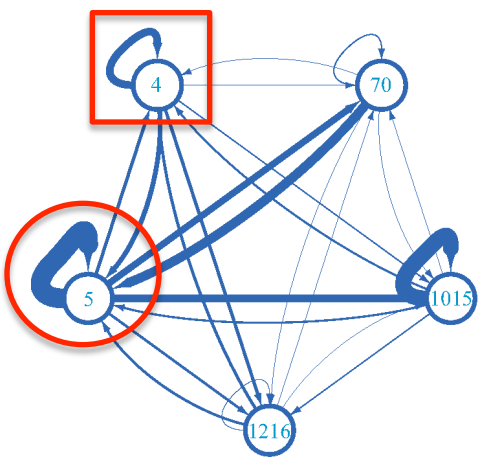
Illustration #2



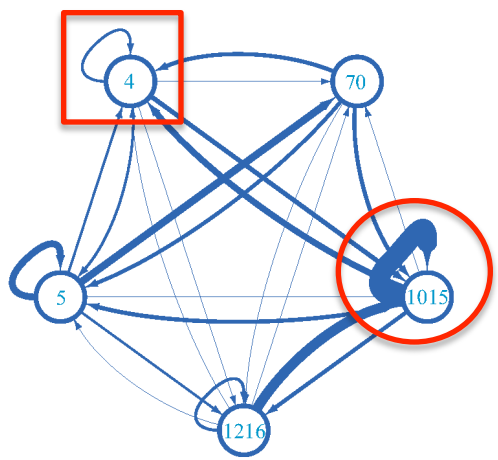
(a) FDN

4: *HiveConf.java*
5: *SemanticAnalyzer.java*
70: *GenMapRedUtils.java*
1015: *ExecDriver.java*
1216: *Driver.java*

A pair of developers in *Hive* committed to a maximum of eight common files when considering the respective top ten most frequently committed files.



(b) FSN-743385



(c) FSN-743435

Bird et al.: *A developer being a minor contributor to a component is partly because he/she is a major contributor to a depending component.*

Bird et al. FSE, 2011

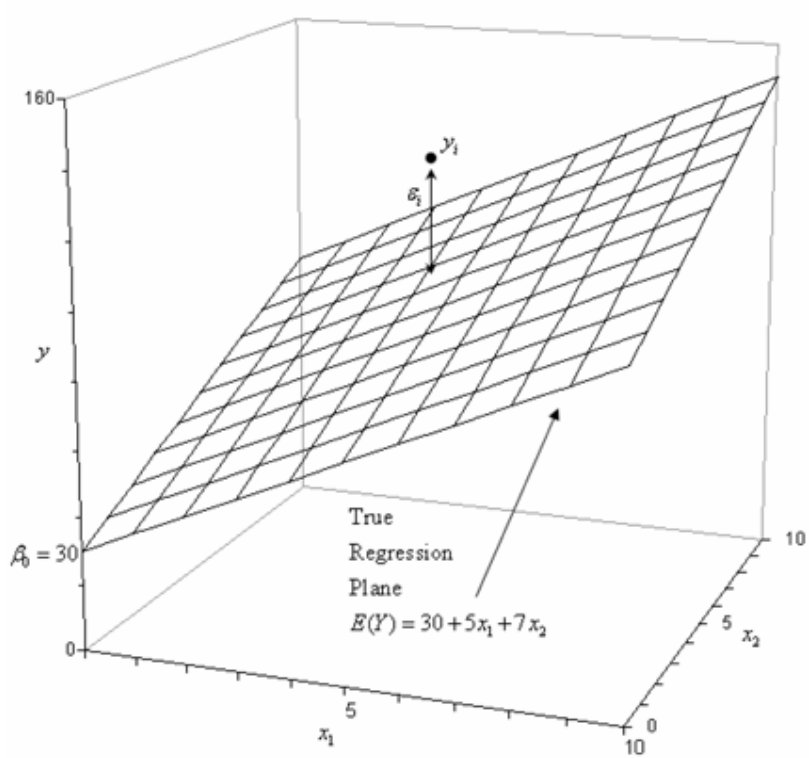
RQ3: Is FSP correlated with code contribution?

Multiple Linear Regression (MLR)

Multiple linear regression model: A linear regression model that contains more than one predictor variable.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon$$

$$Y = 30 + 5x_1 + 7x_2 + \epsilon$$



MLR Model #1

MLR for the code contribution against the congruence

Variable	Estimate	Std. Error	z-value	Pr(>|z|)
(Intercept)	-2.8519	0.9184	-3.105	0.0023
Log(C)	0.6276	0.0680	9.225	4.85e-16
FpC	0.4526	0.0601	7.535	6.16e-12
$Cong$	1.9744	1.1488	1.719	0.0880

Yes! The more productive developers are more likely to shift focus through FDN links.

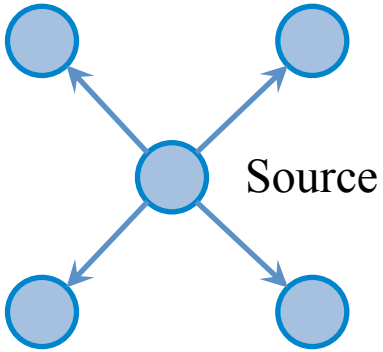
RQ4: Does software structure influence code contribution?

MLR Model #2

MLR for the code contribution against the distributional and structural complexities.

Variable	Estimate	Std. Error	z-value	Pr(>|z|)
(Intercept)	-0.5823	0.5510	-1.057	0.2926
Log(N)	0.6976	0.0636	10.977	<2e-16
P(FSN)	-1.5496	0.2356	-6.578	1.00e-09
Q(FSN)	1.0899	0.1681	6.484	1.61e-09
P(FDN)	-1.1650 −	0.5112	-2.227	0.0243
Q_F(FDN)	0.7555 +	0.4510	1.675	0.0962
Q_B(FDN)	-1.4501 −	0.4972	-2.917	0.0042

- Q_F(FDN)
- Q_B(FDN): More productive developers tend to contribute more to those files (sources) that strongly depend on other files, but less to files (sinks) that others strongly depend upon.



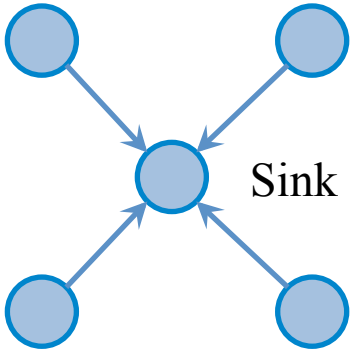


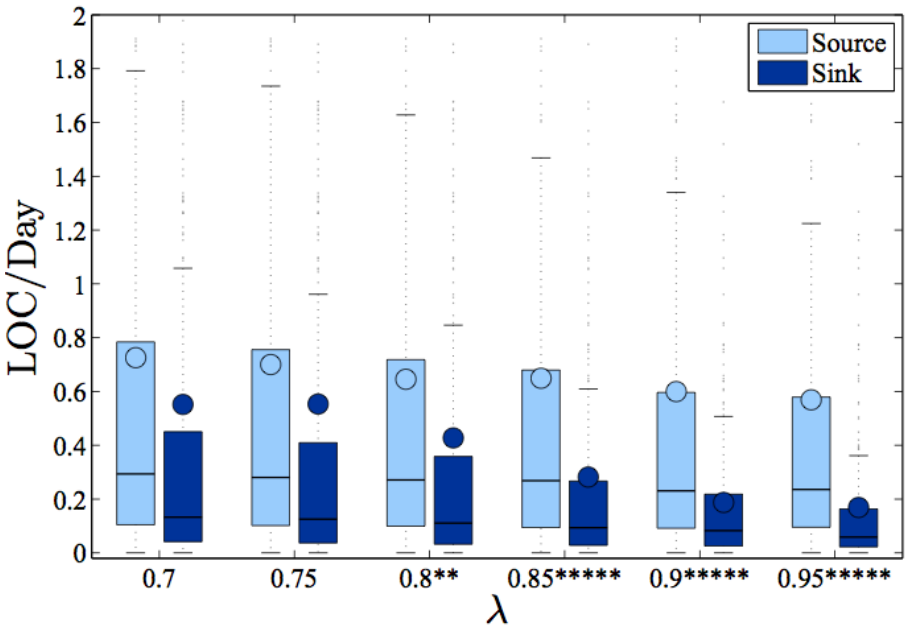
Illustration #3

w_{in}: the sum of weights of incoming links

w_{out}: the sum of weights of outgoing links

Source node: $w_{out} / (w_{out} + w_{in}) > \lambda$

Sink node: $w_{in} / (w_{out} + w_{in}) > \lambda$



Factory in Derby (http://www.oodesign.com/factory-pattern.html):

By adopting the factory pattern, the advantage is obvious: new shapes can be added without changing a single line of code in the framework …… And there are certain factory implementations that *allow adding new products without even modifying the factory class*.

Future Studies

- FSP in class/method level
- Relationship between FSP and defects
- File/class recommendation system
- Optimization of FDN
- Other dependencies between files/classes

Thanks!