Vertical Interaction In Open Software Engineering Communities

Ph.D. Thesis Proposal
Engineering and Public Policy
Computation, Organizations, and Society
Carnegie Mellon University

Patrick Wagstrom

March 5th, 2008

Committee

- James Herbsleb (ISR, co-chair)
- Kathleen Carley (COS/EPP, co-chair)
- Granger Morgan (EPP)
- Audris Mockus (Avaya Labs Research)

Framing The Problem

- Software Engineering has a plethora of development processes
  - XP, Agile, Pair, Scrum, Waterfall, Spiral, RAD, RUP, ...
- Processes differ between companies and within companies
- Participation in Open Source communities further complicates issues
  - New needs to collaborate and share information
  - Suddenly everything is public

Open Source – Changing the Market?

- Open Source Software (OSS) was originally seen as a competitor to commercial software
- Commercial firms readily participate in Open Source projects
  - Alongside both competitors and collaborators
- Most successful Open Source projects have significant commercial involvement
- Many commercial projects include Open Source
- Firms need adapt their processes and learn to communicate and cooperate in these communities

Early Open Source

- Collaboration by independent developers
- Infrastructure provided by project leads
- Little monetary gain
- Licenses were ignorant of commercial use or designed to hinder commercial exploitation

Nascent Commercial Participation

- Into the mid 1990's there was little commercial participation
- IBM really kicked off commercial Open Source
  - Shipped Apache web server
  - Utilized Open Source purely as a commodity
  - Cheaper than developing their own web server
  - Almost purely financial decision

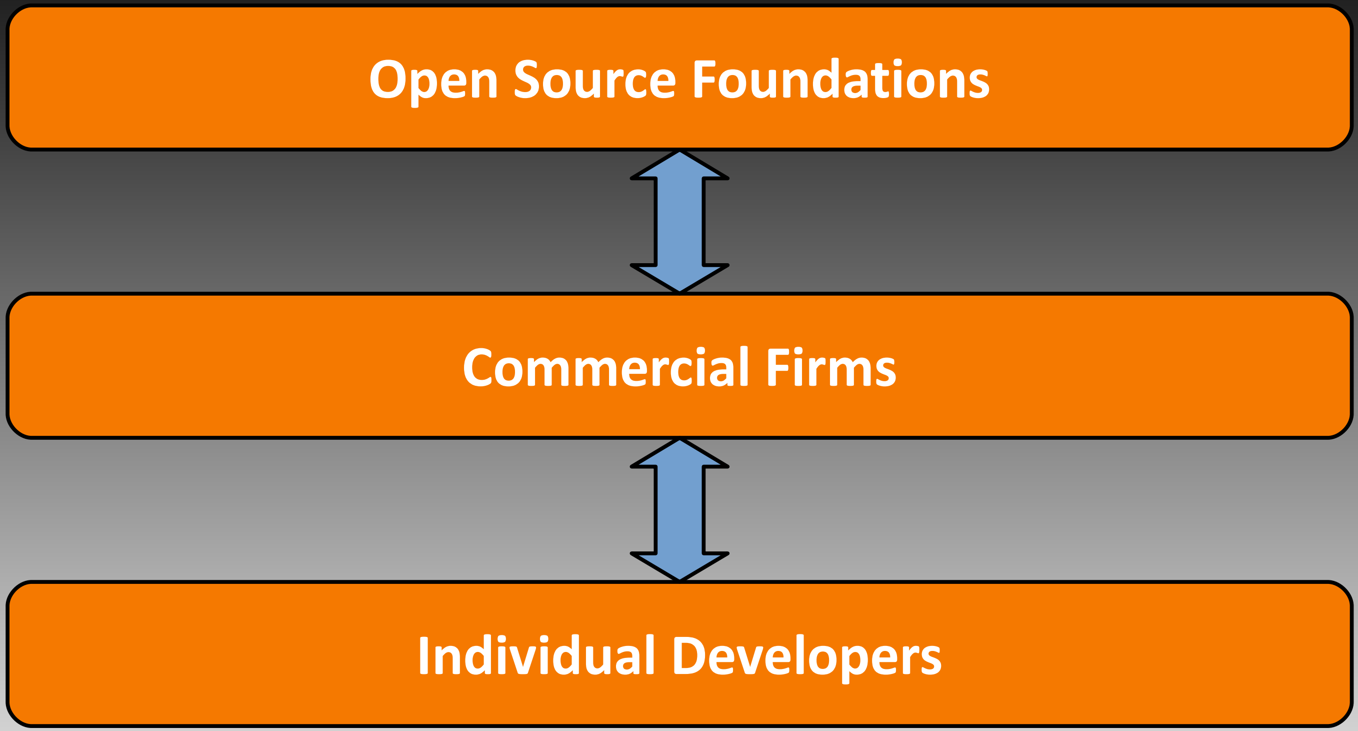
Incorporating Open Source

- Firms next started to include Open Source components into their projects
  - Apple (Mac OS X)
  - Microsoft (NT's TCP/IP)
  - embedded Linux
- Firms were independently leveraging Open Source

Building Communities

- Now firms build and manage entire ecosystems
  - Eclipse, OpenSolaris, Xen
- Primary unit is the firm, not the individual
- Volunteers are scarce – usually university students
- Ecosystems attract previous competitors to rally together
- Launching points for new commercial products

The Structure of Open Source



Open Source Foundations

Commercial Firms

Individual Developers

The Big Problem

- There is academic research on Open Source
  - Most qualitative work addresses only a single firm
  - Most quantitative work doesn't address commercial participation
- Press frequently assumes that OSS is still volunteers working independently
- Huge companies are adopting OSS like strategies in other contexts
  - Boeing is building rockets with an OSS process

The Big Solution

- A vertical examination using two large OSS communities
- Address the realities of commercial participation
- Focus on communication because it's more generalizable across industries
  - Firms and Foundations
  - Firms to Firms
  - Individuals and Firms
  - Individuals to Individuals

§1 – Firms and Foundations in Open Source

- Eclipse has consolidated the IDE market down to two products
- Swarms of former competitors are collaborating on the base technology
- The large market provides great opportunities for new firms to make a name
- Structure of Eclipse allows small firms to have a big impact

The Structure of Eclipse

- **Problem:** The structure is so new, no one knows what is going on
- **Goal:** Develop a comprehensive picture of how firms interact, collaborate, and generate value under the umbrella of a foundation
- **Method:** Qualitative interviews of developers, managers, foundation members, and other affiliated people. Attend annual conference and interview lots more people.

Preliminary Results

- Interviewed ~ 30 individuals from ~ 20 firms
  - Wide breadth of corporate sizes
  - Original Eclipse developers (pre-IBM)
- Assembled a robust history of the project
- Analyzed relationships to Eclipse for 75 firms
- I'm fully buzzword compliant
  - Ask me about my OSGi RCP AJAX client...
- Starting to understand the methods of participation

Preliminary Results

- Identified several business models and incentives for participation
  - Market Consolidation
  - Commodity Utilization
  - Plugin Sales
  - Complimentary Goods
  - Nested Platform Building
  - Customization and Consulting
  - End Users

Potential Problems

- Haven't sufficiently differentiated the business cases
- Not sure how the roles affect decision making in the community
- As outsiders, we could really be missing things

- Luckily, I'm going to EclipseCon in two weeks and presenting to the board of directors

Distinguishing My Contribution

- All technical analysis
- Broad community analysis
- Working with Eclipse foundation to refine story
- Recently, I've been the main person working on this research

§2 – Firm to Firm Interactions

- The foundation performs some key roles, but most of the work still must be done by individual firms
- In the course of our interviews, we gained insight into how firms claim to interact with each other
- Little has been done to create a robust picture of these interactions

Interactions: Translation

- Eclipse ships in a variety of languages
- Most firms benefit from translation as the components are reusable
- But translation is not key element of sales for most firms
- Forces the "Translation Bluffing Game"
- IBM usually caves and does the translations
  - Highly centralized

Interactions: SWT

- Eclipse uses a widget set called SWT
- Originally was IBM specific
- Later generalized into a new Java toolkit
- Firms that want a new widget must write it themselves
- Widgets are generally independent
  - Highly distributed

Interactions: Editor

- Text editor is the primary interaction tool in Eclipse
- Key example of a commodity technology
- Utilized in many commercial IDEs based on Eclipse
- Each firm has small customizations
- Usually contributes code back to the common component
  - Highly collaborative

Understanding Collaboration

- **Problem:** Firms collaborate on components in Eclipse, but no one is certain of the "big picture"
- **Goal:** A quantitative overview of contributions to Eclipse components by firm
- **Method:** Identify contributors to Eclipse source code by firm and then examine the contributions of each firm to components in Eclipse

Modeling Interactions of Firms

- **Problem:** Firms collaborate over channels other than source code. These channels have multiple possible representations.
- **Goal:** Understand the implications of assumptions in generating networks from archival data
- **Method:** Generate many different networks using different techniques and compare what the results mean for position

“True” Interaction Models In Eclipse

- **Problem:** We have no idea how truly collaborative Eclipse is
- **Goal:** Generate a network structure that is backed with explanations of possible variance
- **Method:** Utilize earlier network formulations to create a overall picture of the participation in Eclipse. Compare this network to data about collaboration from interviews and analysis in §1

Possible Issues

- Data collection
  - I have bug data, but no information on developers, need to spider the data
  - Identification of firms requires use of work email addresses. IP licensing agreement strongly recommends but does not require use of work email. May be possible to get access to some info from Eclipse Foundation.
  - The web accessible Eclipse mailing lists have email addresses sanitized
- Determination of “best” network model

§3 – Individual and Firm Interactions

- **Problem:** Not all OSS communities are commercial. Commercial firms entering these communities have the potential to disrupt the community.
- **Goal:** Understand how commercial participation affects subsequent volunteer participation.
- **Method:** Longitudinal multi-level analysis of the GNOME project identifying the impact of commercial developers on volunteer participation.

There Goes the Neighborhood

- Two part study
- 18 developer interviews to understand developer motivations, viewpoints, and opinions of commercial firms
- Quantitatively test:
  - Cognitive complexity Issues
  - Volunteer developer signaling and project momentum
  - Heterogeneity in developer populations
  - Clash of norms and values

Results

- Cognitive complexity not an issue
- Signaling and momentum are supported
- Heterogeneity is not supported
- Differences of norms and values is supported
  - Community focused firms attract volunteer developers
  - Product focused firms have no statistically significant relation

Proposed Work – Signaling

- **Problem:** Unable to differentiate between signaling and momentum as cause for increased volunteer participation
- **Goal:** Test if volunteers preferentially communicate with commercial firms that may hire them
- **Method:** Generate networks of email messages in the community and test if volunteers preferentially communicate with commercial developers

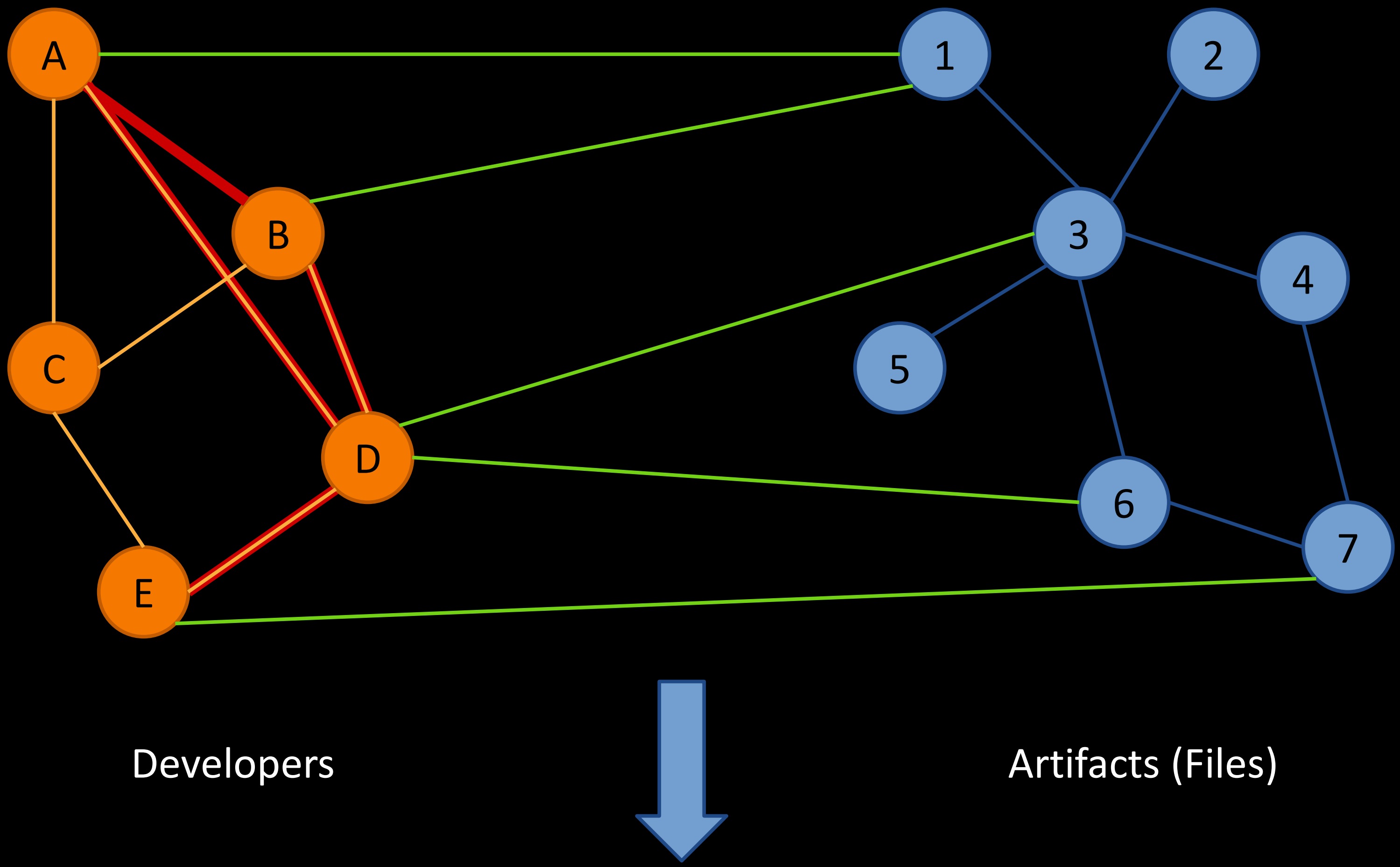
Proposed Work – Feature Preferences

- **Problem:** Interviews indicate some preference for corporations that work on features useful to volunteers
- **Goal:** Empirically test if new volunteers preferentially work on features they find useful
- **Method:** For a selection of projects, identify features and cluster networks from CVS and Bugzilla to identify "hot spots" of new volunteers

§4 – Individual to Individual Interactions

- Firms can exert a lot of control over employees, but in the end, people make their own decisions
- Developers need to choose who to interact with
- Must ensure that technical dependencies are accounted for in communication

Socio-Technical Congruence



0.67 Overall Congruence

Individualized Congruence

- **Problem:** Tools are being developed for STC, but isn't clear how individuals affect STC
- **Goal:** Develop a metric for STC that addresses the actions of individuals
- **Method:** Subdivide communication and dependencies into ego networks. Create a weighted coordination requirements network to evaluate if information was properly directed

Preliminary Work

- Created two metrics: Unweighted (UIC) and Weighted Individual Congruence (WIC)
- Analyzed approximately 8,000 bugs from 10 projects in GNOME
- More communication decreases performance
- More coordination requirements increases performance
- **Key Question:** Are individualized STC and overall STC just new proxies for centrality related metrics?

Uncertainty Analysis

- **Problem:** Network methods often have non-linear responses. We also have uncertainty about the underlying network structure.
- **Goal:** understand what effect errors of omission and commission have on STC
- **Method:** Monte Carlo to create response surface for a variety of networks of different densities. Farm computing out to Amazon EC2.

Uncertainty Analysis

- **Problem:** Most communication in STC metrics is from archives and it is not known if the communication was actually relevant
- **Goal:** Create a set of probabilistic metrics for observed communication in STC
- **Method:** Create distribution of probabilities for edges in $\boldsymbol{C_A}$. Probabilistically instantiate actual communication network. Provides a set of confidence bounds for STC.

Thesis Impact – Foundations

- Provide guidance in recruiting firms
- Better develop standards for cooperation and collaboration
  - Particularly regarding how firms work together
- Understand collaboration and direct new projects accordingly

Thesis Impact – Firms

- Method for analyzing an ecosystem
  - Understand roles of competitors, collaborators
- Understand the required resource contribution
- Participate in a manner that doesn't disrupt the community

Thesis Impact – Individuals

- Understanding of commercial firms in Open Source
  - They're not the enemy
- Improved metrics for collaborative tools
  - Know who to communicate with

Timeline

March

- Submit Corporate Involvement paper to ISR (§3)
- Spider Eclipse Bugzilla Profiles (§2)
- Retool and update R scripts for Congruence (§4)
- Present at EclipseCon (§1)
- Schedule and begin followup interviews (§1,2)

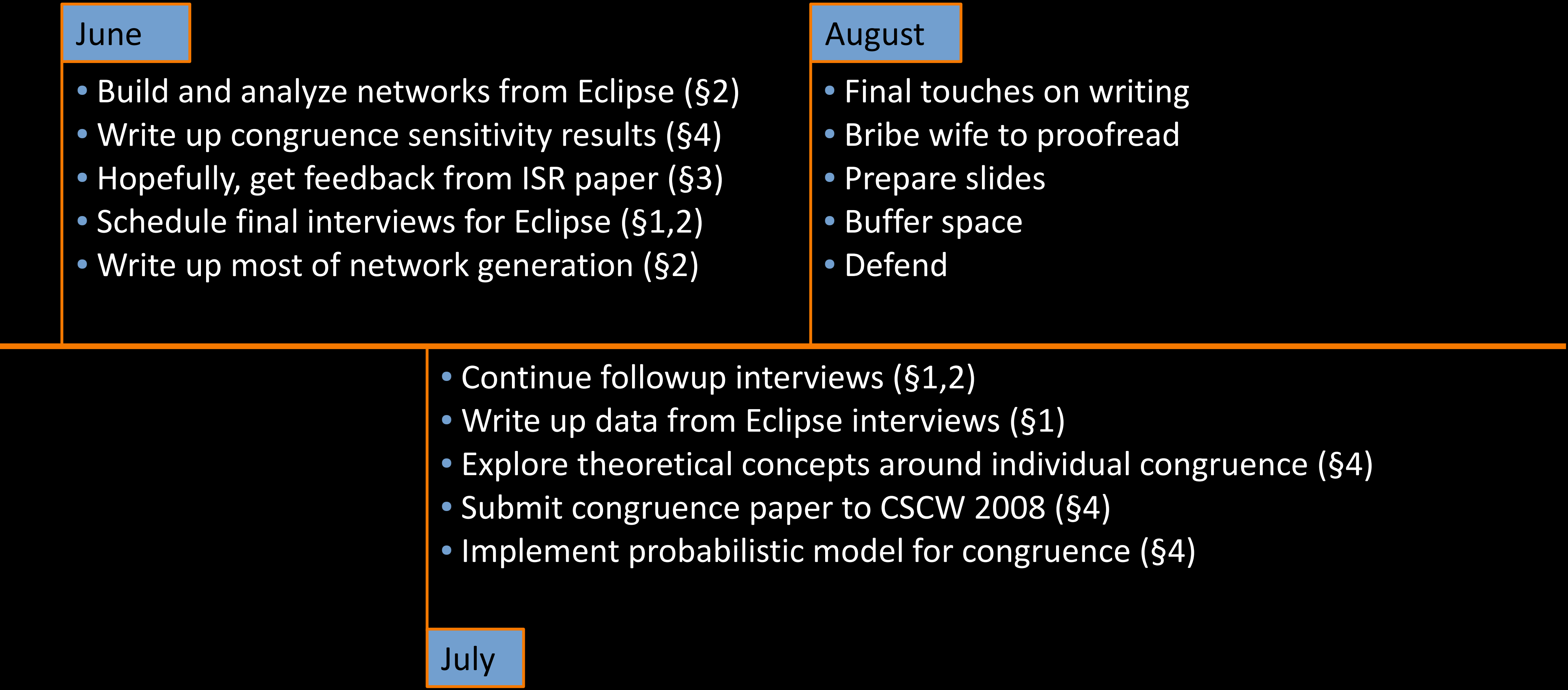
April

- Load and Clean up Data from Eclipse (§2)
- Explore theoretical concepts around individual congruence (§4)
- Submit congruence paper to CSCW 2008 (§4)
- Implement probabilistic model for congruence (§4)
- Continue followup interviews (§1,2)

May

- Sloan Industry Studies Conference (§1,2)
- Analyze Probabilistic Model (§4)
- STC 2008 (§4)
- Code methods to generate Eclipse networks (§2)
- Incorporate feedback from STC and Sloan (§1,4)
- Affinity networks (§3)

Timeline



June

- Build and analyze networks from Eclipse (§2)
- Write up congruence sensitivity results (§4)
- Hopefully, get feedback from ISR paper (§3)
- Schedule final interviews for Eclipse (§1,2)
- Write up most of network generation (§2)

July

- Continue followup interviews (§1,2)
- Write up data from Eclipse interviews (§1)
- Explore theoretical concepts around individual congruence (§4)
- Submit congruence paper to CSCW 2008 (§4)
- Implement probabilistic model for congruence (§4)

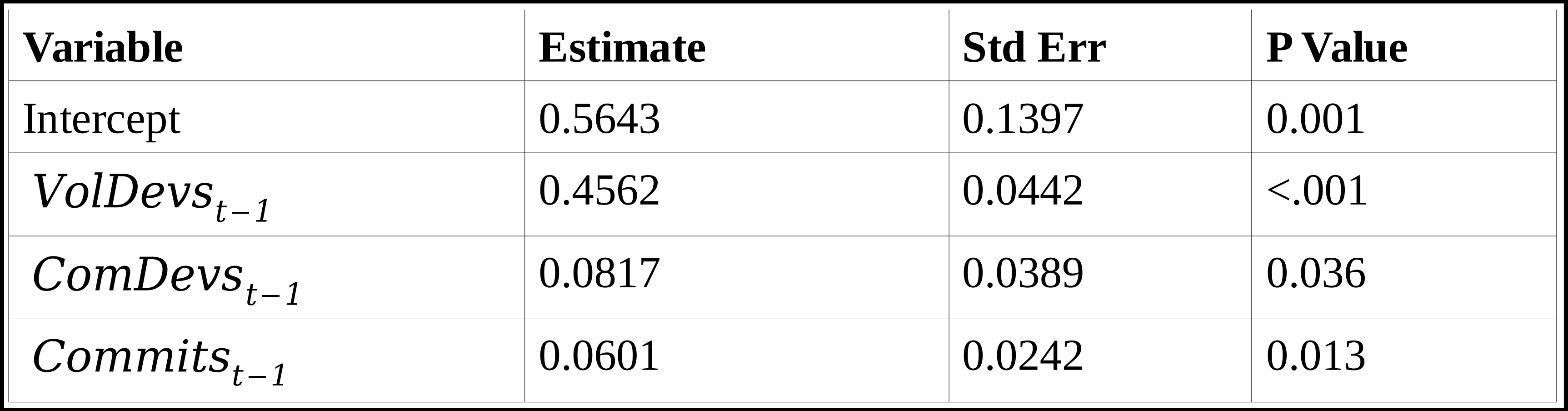
August

- Final touches on writing
- Bribe wife to proofread
- Prepare slides
- Buffer space
- Defend

End of Presentation

There Goes the Neighborhood

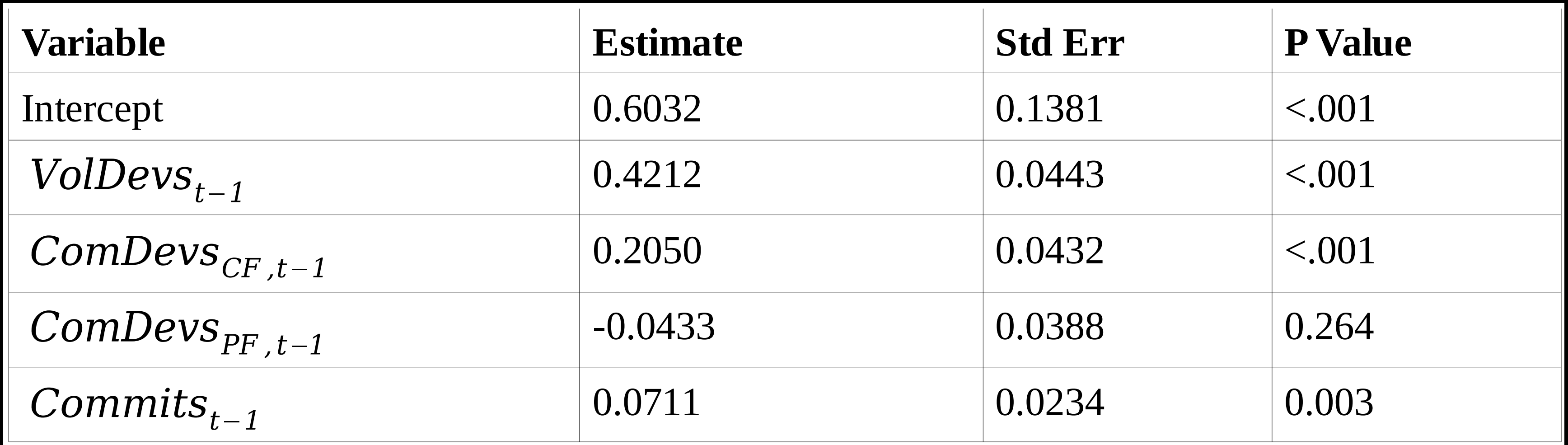
- Momentum and Signaling – Project Level



Variable	Estimate	Std Err	P Value
Intercept	0.5643	0.1397	0.001
$VolDevs_{t-1}$	0.4562	0.0442	<.001
$ComDevs_{t-1}$	0.0817	0.0389	0.036
$Commits_{t-1}$	0.0601	0.0242	0.013

There Goes the Neighborhood

- Norms and Values – Project Level



Variable	Estimate	Std Err	P Value
Intercept	0.6032	0.1381	<.001
$VolDevs_{t-1}$	0.4212	0.0443	<.001
$ComDevs_{CF,t-1}$	0.2050	0.0432	<.001
$ComDevs_{PF,t-1}$	-0.0433	0.0388	0.264
$Commits_{t-1}$	0.0711	0.0234	0.003

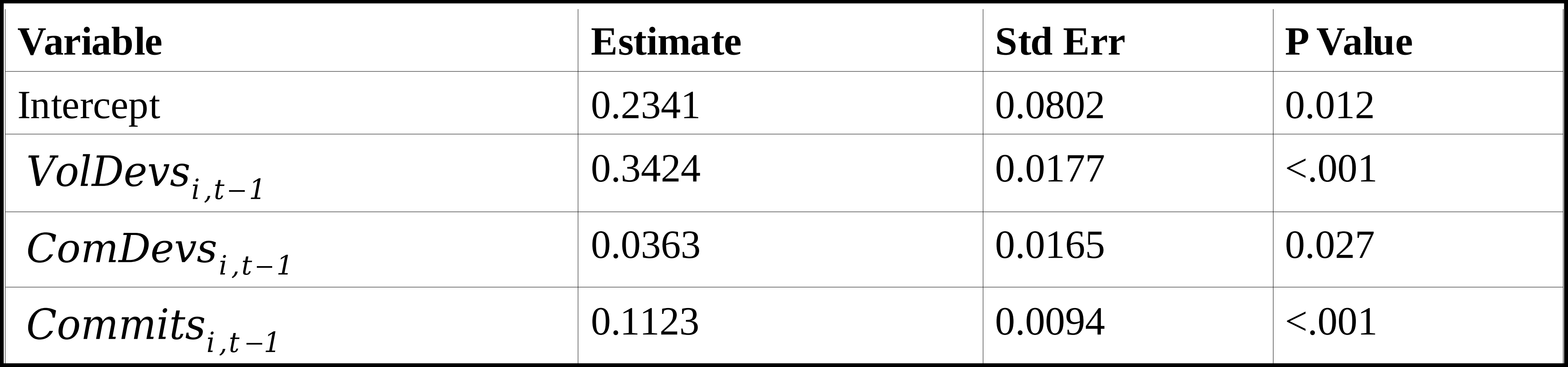
There Goes the Neighborhood

- Mediating Differences

Variable	Estimate	Std Err	P Value
Intercept	0.6122	0.1387	<.001
$VolDevs_{i,t-1}$	0.4527	0.0471	<.001
$ComDevs_{CF,i,t-1}$	0.2165	0.0453	<.001
$ComDevs_{PF,i,t-1}$	-0.0177	0.0437	0.685
$Commits_{i,t-1}$	0.0939	0.0247	<.001
$BugProjects_{i,t-1}$	-0.0030	0.0001	0.046
$DevMailMessages_{i,t-1}$	0.00005	0.0001	0.692
$CVSProjects_{i,t}$	-0.0028	0.0012	0.025

There Goes the Neighborhood

- Cognitive Load – Module Level



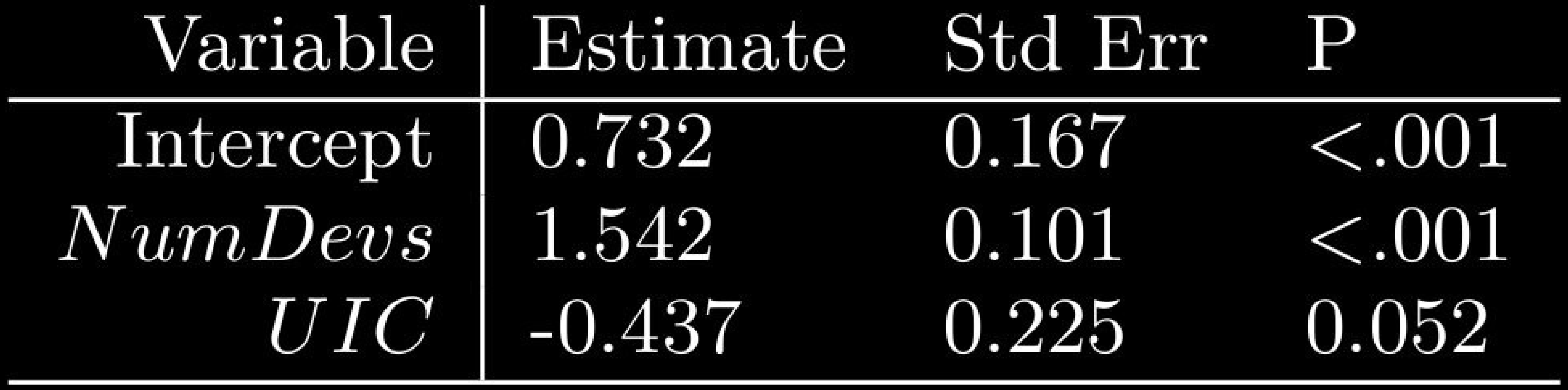
Variable	Estimate	Std Err	P Value
Intercept	0.2341	0.0802	0.012
$VolDevs_{i,t-1}$	0.3424	0.0177	<.001
$ComDevs_{i,t-1}$	0.0363	0.0165	0.027
$Commits_{i,t-1}$	0.1123	0.0094	<.001

Individualized Congruence Formulas

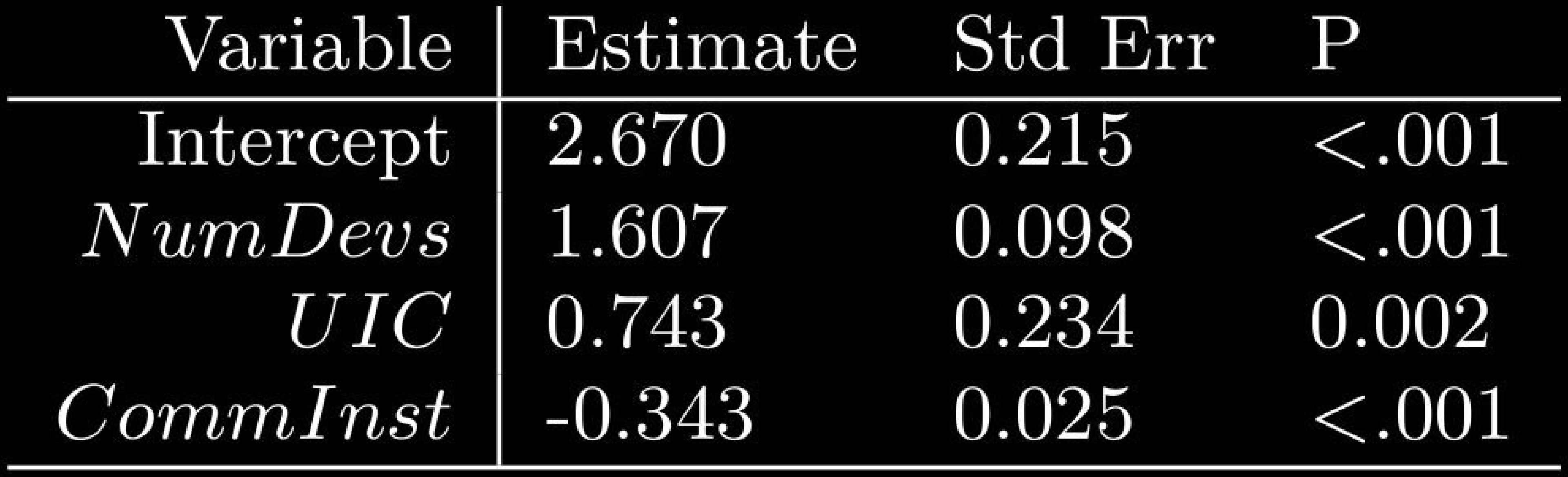
$$UIC_i = \frac{\sum (\mathbf{C}_\mathbf{R}[i,] \times \mathbf{C}_\mathbf{A}[i,]) + \sum (\mathbf{C}_\mathbf{R}[,i] \times \mathbf{C}_\mathbf{A}[,i])}{\sum \mathbf{C}_\mathbf{R}[i,] + \sum \mathbf{C}_\mathbf{R}[,i]}$$

$$WIC_i = \frac{\sum \left(\mathbf{C}_\mathbf{R}\left[i,\right] \times d\left(\mathbf{C}_\mathbf{A}\left[i,\right]\right)\right) + \sum \left(\mathbf{C}_\mathbf{R}\left[,i\right] \times d\left(\mathbf{C}_\mathbf{A}\left[,i\right]\right)\right)}{\sum d(\mathbf{C}_\mathbf{R}\left[i,\right]) + \sum d(\mathbf{C}_\mathbf{R}\left[,i\right])}$$

UIC: Preliminary Results

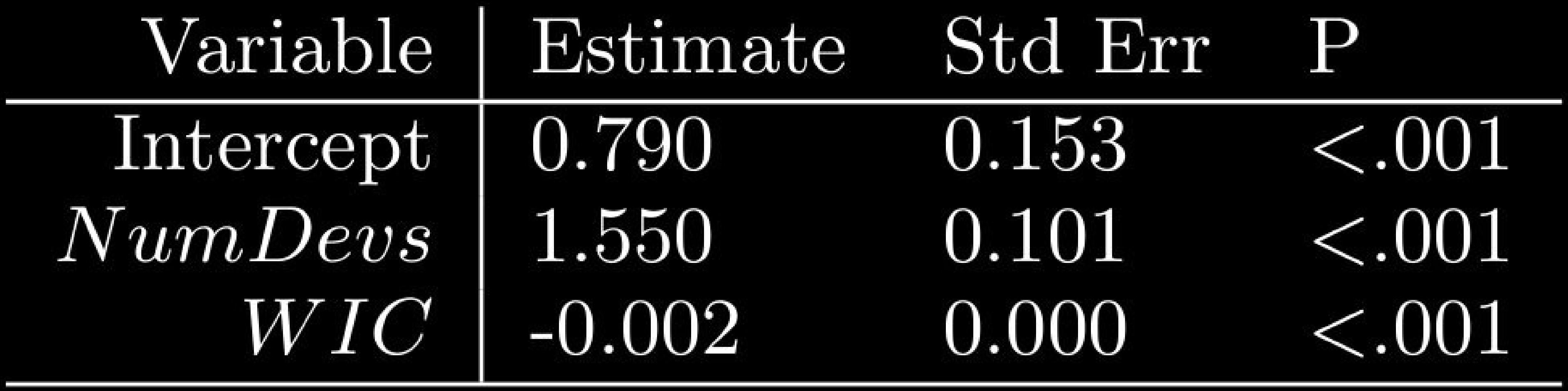


Variable	Estimate	Std Err	P
Intercept	0.732	0.167	<.001
$NumDevs$	1.542	0.101	<.001
UIC	-0.437	0.225	0.052

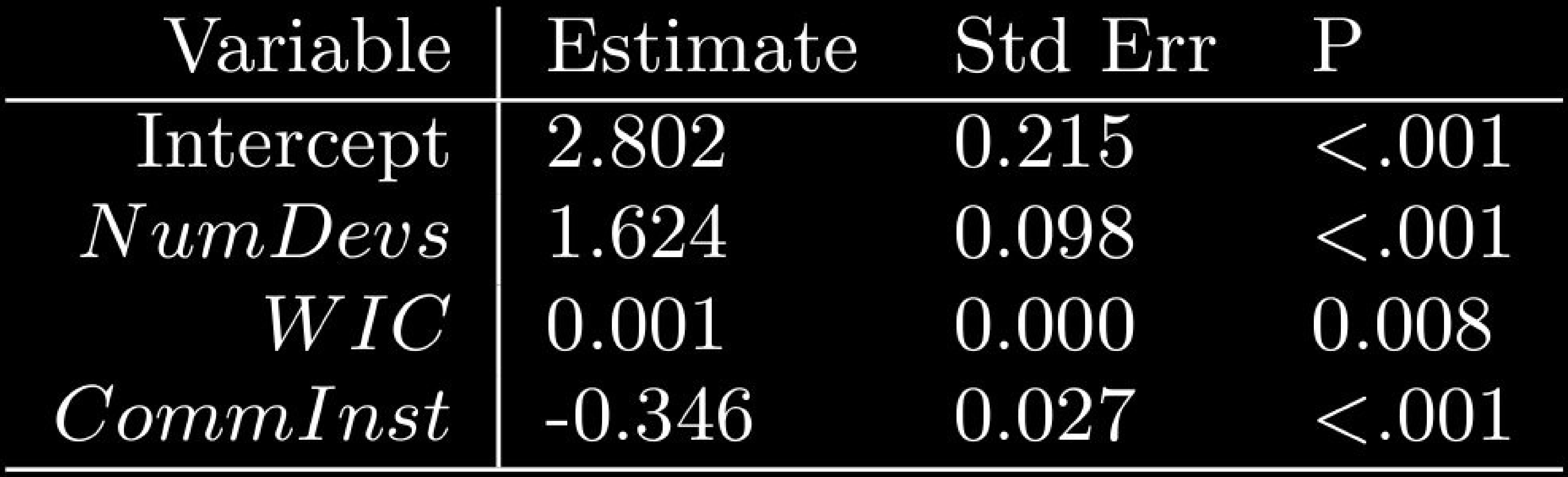


Variable	Estimate	Std Err	P
Intercept	2.670	0.215	<.001
$NumDevs$	1.607	0.098	<.001
UIC	0.743	0.234	0.002
$CommInst$	-0.343	0.025	<.001

WIC Preliminary Results



Variable	Estimate	Std Err	P
Intercept	0.790	0.153	<.001
$NumDevs$	1.550	0.101	<.001
WIC	-0.002	0.000	<.001



Variable	Estimate	Std Err	P
Intercept	2.802	0.215	<.001
$NumDevs$	1.624	0.098	<.001
WIC	0.001	0.000	0.008
$CommInst$	-0.346	0.027	<.001