System Perspective and Lean Coordination

The case of Open Source Software

Joao Castro

Committee: Warren Seering
Eric Rebentisch
Chris Magee

Coordination and lean

What they mean to me:

- Lean as being effective, efficient, minimal waste

- Coordination as being effective in managing interactions in complex environments

Can we get efficient coordination?

- In high complexity environments?

Product development is the study context

Coordination

- Coordination is consistently cited as one of the most important factors of competitive advantage

> "The primary task of management is to get people to **work together in a systematic way**"
>
> Christensen et al, (2006), *The Tools of Cooperation and Change.* Harvard Business Review

- Literature describes many methods
  - Centralize people
  - Centralize information
  - Facilitate Communication
  - Structure Communication
  - Structure Processes

Field exploration

- Visited and collected data, interviews from different PD companies:
  - Industrial machinery (2)
  - Aerospace (2)
  - Medical equipment
  - Services (2)
  - Food packaging
  - PD consultancy
- Factors affecting coordination are varied:
  - Team size
  - Schedule
  - Reputation for prob. solving
  - Product complexity

Where to focus?

> "Ohno thought that assembly workers could probably do most of the functions of the specialists and **do them much better because of their direct acquaintance with conditions on the line.**"
>
> Womack, Jones, and Roos, (1990)
>
> *The Machine That Changed The World*

Where to focus?

- Brook's Law implies that the **ideal size for a programming team is one** – a single developer who never has to stop to communicate with a colleague.

- This approach streamlines everything, and it also provides insurance that the project will retain **"conceptual integrity"**.

Rosenberg, (2007), *Dreaming in Code*.

Where to focus?

"I also believed there was a **certain critical complexity above which a more centralized, a priori approach was required.**"

"Linus Torvalds's style of development came as a surprise. (…)The fact that this bazaar style **seemed to work, and work well**, came as a distinct shock."

Eric Raymond, (1999) ***The Cathedral and the Bazaar***

Where to focus?

But, as Galileo is said to have murmured after officially recanting his statement that the earth moves around the sun:

"And yet it moves!"

What is going on here?"

von Krogh and von Hippel (2006)

The Promise of Research on Open Source Software

Driving Hypothesis

Individuals in a collaborative environment and behaving autonomously are able to ***efficiently*** solve complex problems

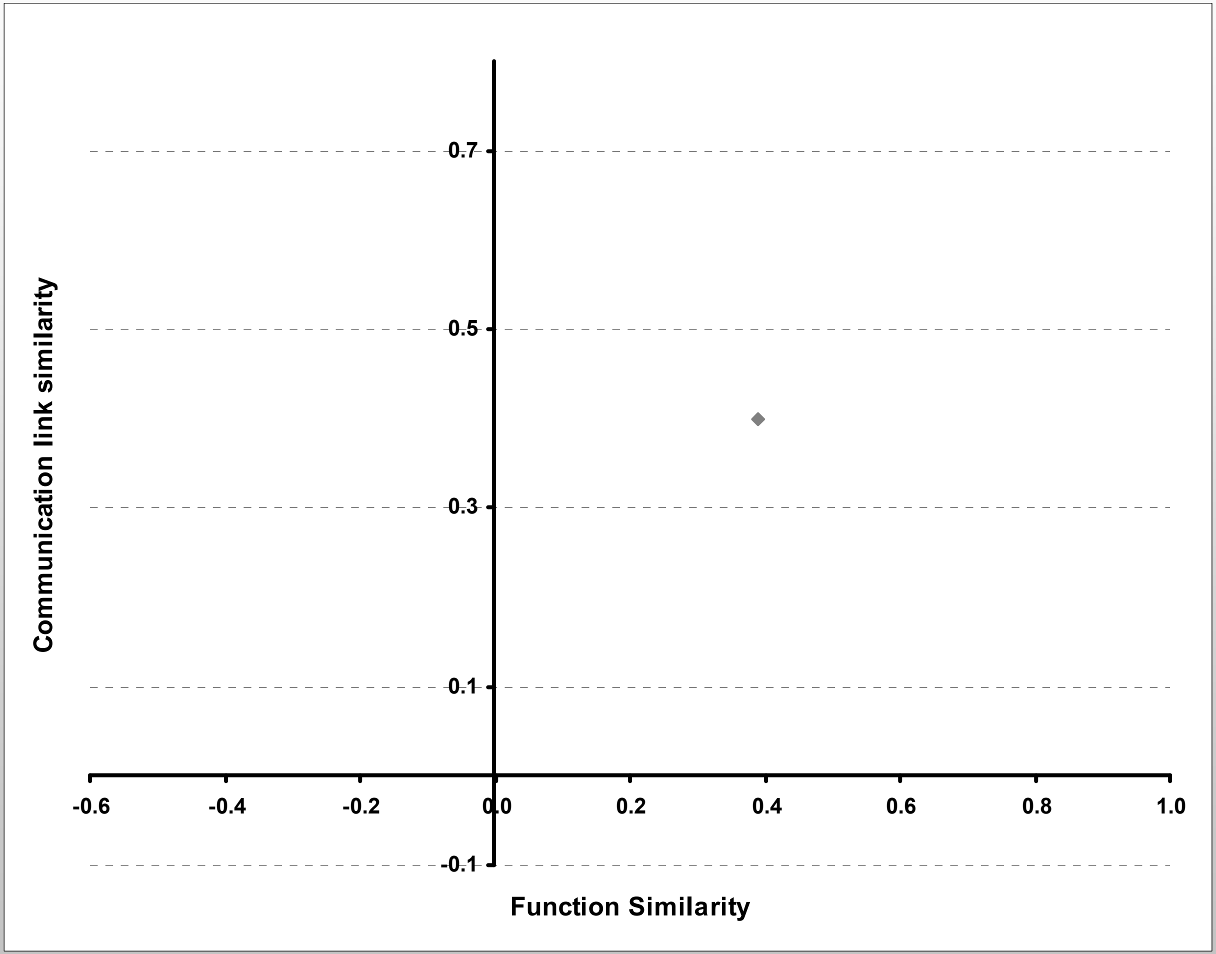
In other words:

> Coordination is possible without heavy supervisory and overhead methods.

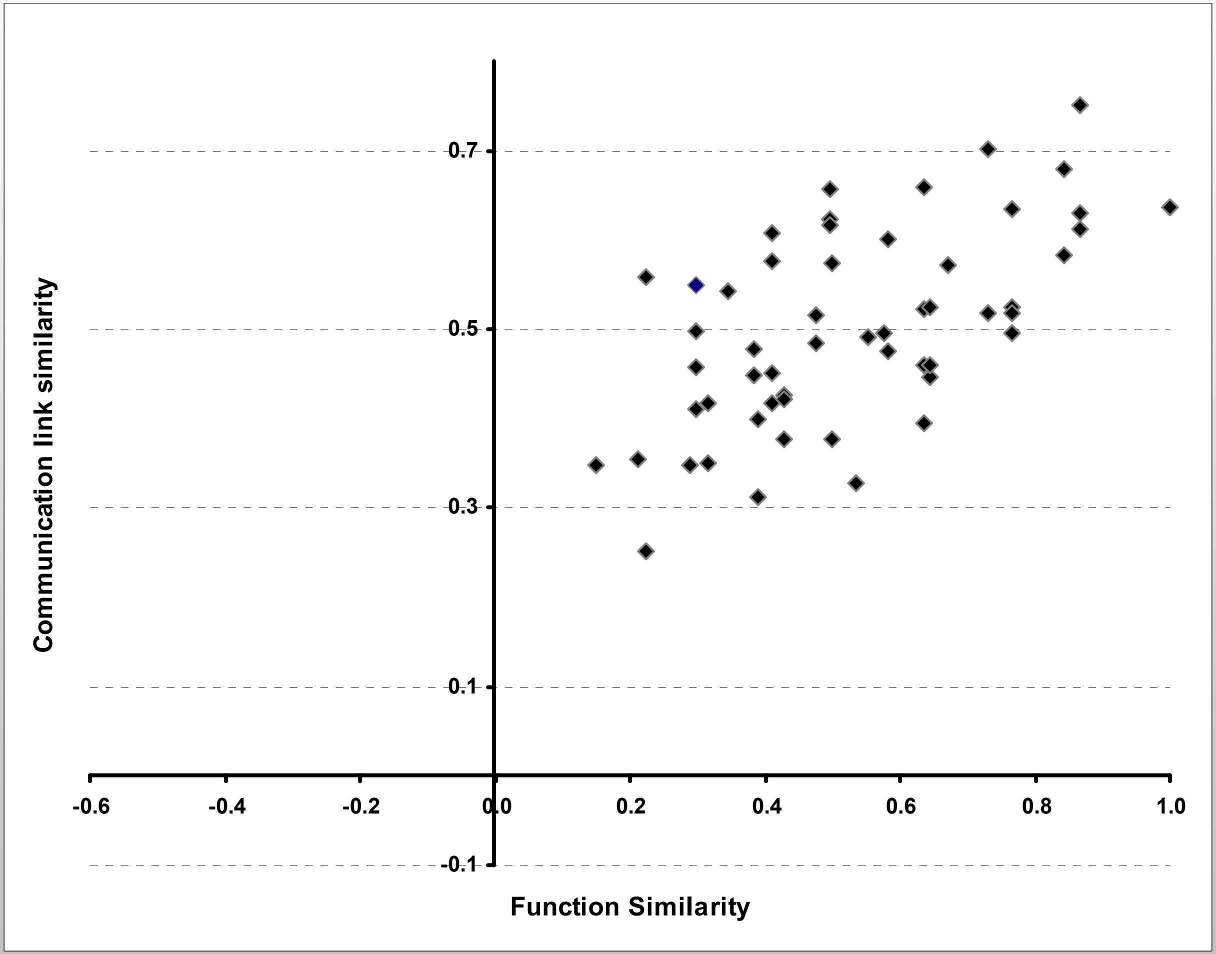
Is there a driver behind what connections are made?

- Connections in a complex project will happen
- Given two similar projects, will the connections also be similar?
- Test scenario:
  - Multiple concurrent engineering sessions
    - No barriers to communication in each session
  - Sessions have different objectives and different function areas are selected and staffed

Are connections made on purpose?

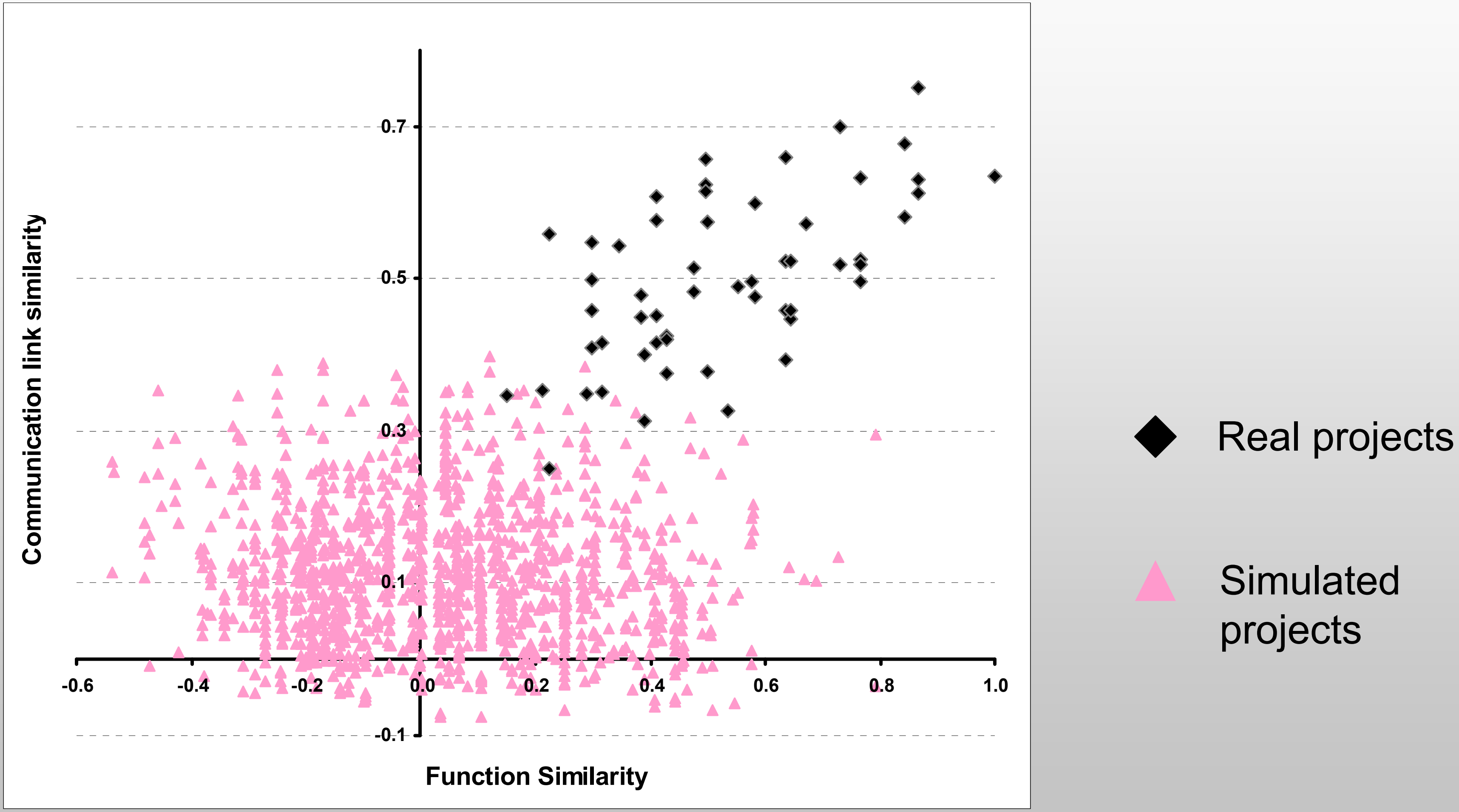


Are connections made on purpose?



Data for real projects provided by Mark Avnet

Are connections made on purpose?



Data for real projects provided by Mark Avnet

Hypotheses and questions

- Hx: Different levels of product complexity require different patterns of people communication. Number, focus and attention span vary.
- Hx: Level of oversight by a person is extremely limited (they only see what they do) when compared to the whole system.
- Hx: People's attention (as verified by their footprint) varies through time. Once something is done, they move on and do not return to it.
- Hx: Systems that operate under the freedom of participants have high redundancy communication channels.
- Hx: System critical components are verified by several people at different times.

Finding more data

- Product complexity, component interaction and specialty interaction is context specific and varies across industries

- But, at a low-level, we can abstract to:
  - Component A <- logical interaction - > Component B

- This allows us to study the same problem in different industries and try to understand and generalize

- But to understand different behaviors, a lot of very detailed data from several projects is required
  - **Focus on one area: software**

Why software? Fits the topic

- Software code is also made of a set of logical interactions:
  - procedures, functions, variables and objects
- Complex software is developed collaboratively by teams.
  - Each member works on a sub-part of the system that interacts
  - Members often work on code written by others
  - Teams can typically be geographically dispersed
- Software is key part in *almost* all modern complex products

Why software? Good data for research

- Software engineering practices have excellent book-keeping methods which give us:
  - Fine grained information
  - Complete information. Long history on past projects is available
  - Uniform data over time
  - Even small projects generate large volumes of changes making it possible to detect even small effects statistically
- The data collection is nonintrusive / non-disturbing
  - doesn't require resources from project to help with the data collection
- The data collection is cheap
  - no impact on the project as this data collection is already performed

Adapted from (Mockus, Weiss et al. 2003)

- Data is ripe for processing
  - Using computer to process and analyze a lot of information

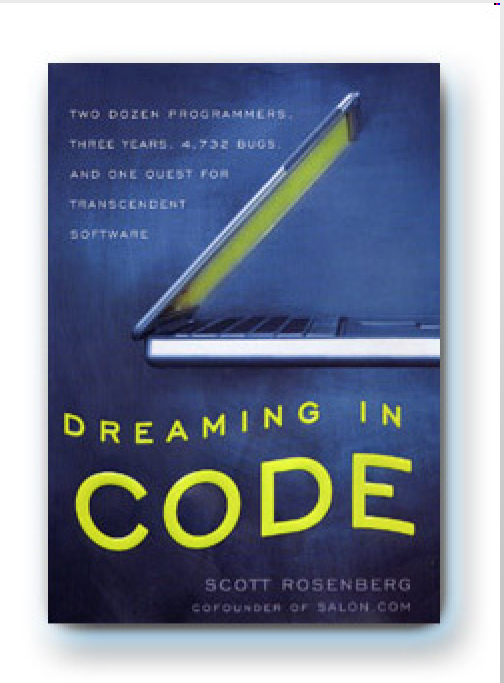
Why software? Even better data

- The information-based nature of software products brings another benefit in that **we can track the evolution of a design over time**. (…) For a researcher, this presents an opportunity to follow the “living history” of a design, a technique that is typically not possible for physical products.

  Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code (MacCormack, Rusnack, Baldwin 2005)

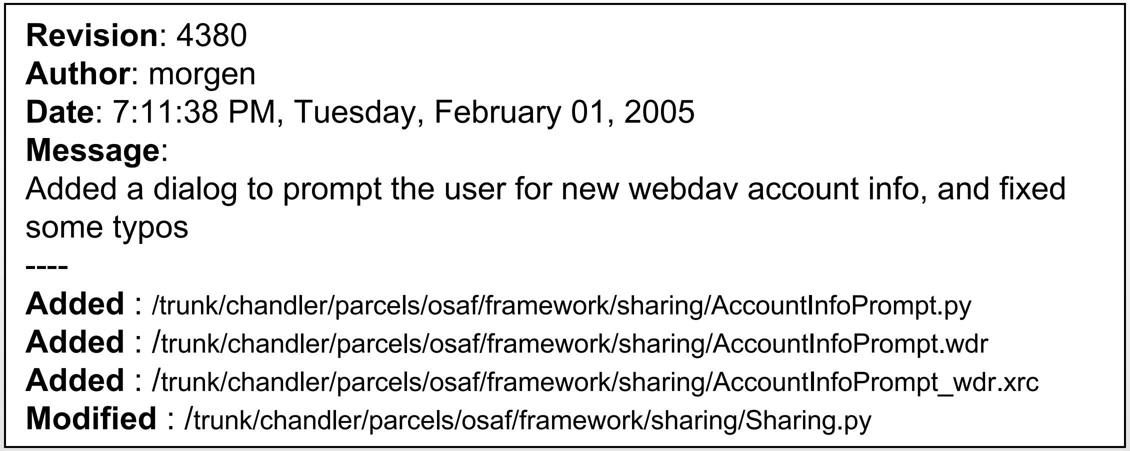
Why software? Available literature

- Academic Research
  - Software development process
  - Coordination in software projects
  - Measures of software complexity
  - Visualization of software and team participation
  - Social settings, motivations and behaviors of participants
  - Social network analysis of software projects
  - System evolution
  - Case studies
- Ethnographies


TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE
DREAMING IN CODE
SCOTT ROSENBERG
COFOUNDER OF SALON.COM

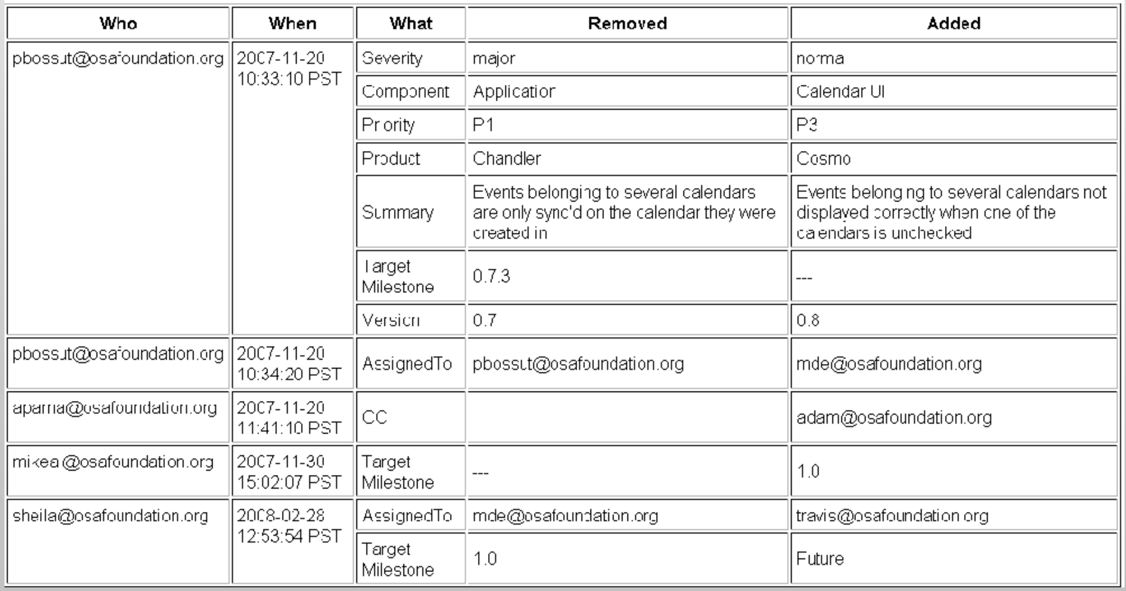
Software Data. What it looks like

- Change in the code



Revision: 4380
Author: morgen
Date: 7:11:38 PM, Tuesday, February 01, 2005
Message:
Added a dialog to prompt the user for new webdav account info, and fixed some typos
----
Added : /trunk/chandler/parcels/osaf/framework/sharing/AccountInfoPrompt.py
Added : /trunk/chandler/parcels/osaf/framework/sharing/AccountInfoPrompt.wdr
Added : /trunk/chandler/parcels/osaf/framework/sharing/AccountInfoPrompt_wdr.xrc
Modified : /trunk/chandler/parcels/osaf/framework/sharing/Sharing.py

- Bug correction activity



Who	When	What	Removed	Added
pbossut@osafoundation.org	2007-11-20 10:33:10 PST	Severity	major	normal
		Component	Application	Calendar UI
		Priority	P1	P3
		Product	Chandler	Cosmo
		Summary	Events belonging to several calendars are only sync'd on the calendar they were created in	Events belonging to several calendars not displayed correctly when one of the calendars is unchecked
		Target Milestone	0.7.3	---
		Version	0.7	0.8
pbossut@osafoundation.org	2007-11-20 10:34:20 PST	AssignedTo	pbossut@osafoundation.org	mde@osafoundation.org
aparna@osafoundation.org	2007-11-20 11:41:10 PST	CC		adam@osafoundation.org
mikeal@osafoundation.org	2007-11-30 15:02:07 PST	Target Milestone	---	1.0
sheila@osafoundation.org	2008-02-28 12:53:54 PST	AssignedTo	mde@osafoundation.org	travis@osafoundation.org
		Target Milestone	1.0	Future

Projects of interest - Open Source

- Open Source Software is a type of software project that relies on a loose articulation between developers.

- Open source software projects are based on voluntary contributions and **involve only very light coordination** activities by a central project team

  Kogut and Metiu (2001) Open-Source Software Development and Distributed Innovation

- "What is perhaps most surprising about the process is that it **lacks many of the traditional mechanisms used to coordinate** software development, such as plans, system-level design, schedules, and defined processes."

  Mockus, Fielding et al. (2002) Two Case Studies of OSS Development: Apache and Mozilla

- "everyone, under this type of project management, is self-determining"

  Mockus and Herbsleb (2002) Why Not Improve Coordination in Distributed Software Development by Stealing Good Ideas from Open Source?

Open Source Software

- Because of their open policies, project data is already public
  - Time to gain access to projects is cut to almost zero
  - Data is available online
  - No need to travel
  - No need to navigate NDAs

Rewriting our Hypotheses and questions

- Hx: Different levels of product complexity require different patterns of people behavior. Number, focus and attention span vary.
- Hx: Level of oversight in code by a person is limited (the files they edit mostly reference themselves)
- Hx: Does people's attention (as verified by their footprint) varies through time. Do they come back to their older files while editing new ones?
- Hx: How well does the ensemble of perspectives cover the whole code?
- Hx: A developer engages in coordination only with those who are part of his system view
- Hx: Most time is spent on the boundary components than on independent components.

Projects collected (so far)

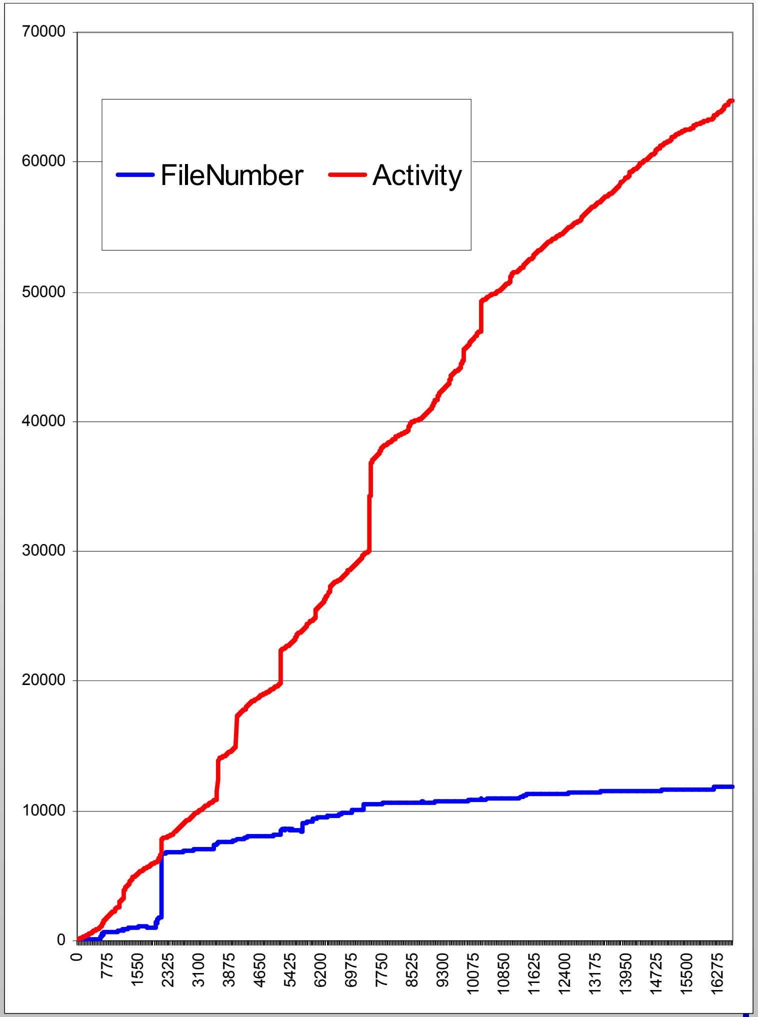
Project name	Description	Number of developers	Got code	Source code files	Lines of code	Number of functions	Number of edits in DB	Got Bugs	Number of bugs	Got ML	Total # of Msgs	Got IRC
Chandler	PIM	44	Yes	5374	1195429	2560	14835	Yes	12427	Yes	10405	Can get
Audacity	Audio editing		Can get					No				
Apache	Web Server	95	Yes	8645	1382475	694	18020					
Wireshark		40	Yes	12560	5800442	7374	26421	Can get				
MythTV	Media center	41	Yes	20792	3941132	18091	17221	Can get				
Rsync	Backup		Yes	382	73392	218		Can get		Can get		
Git	Repository mgmt		Yes	1796	322768	1042				Can get		
Android	Mobile OS		Very big									
GNUmeric	Spreadsheet	217	Yes	6583	4818253	2366	15827					
Gimp	Image processing	260	Yes	14455	6648419	7610	24772					
Juice	Podcast		No code									
Songbird		30	Yes	14433	1924184	1626	7954					

825 859 records in database so far

Example of data from an open source project

Chandler

- Tracked from Aug’02 to Nov’08
- 43 developers
- Source Code:
  - 14 835 changes (commits)
  - 5 347 files
  - 2 560 functions, 1 195 429 loc
- Bugs:
  - 12 427
- Mailing list:
  - 10 405 emails exchanged
- Internet chat:
  - ?



Data analysis

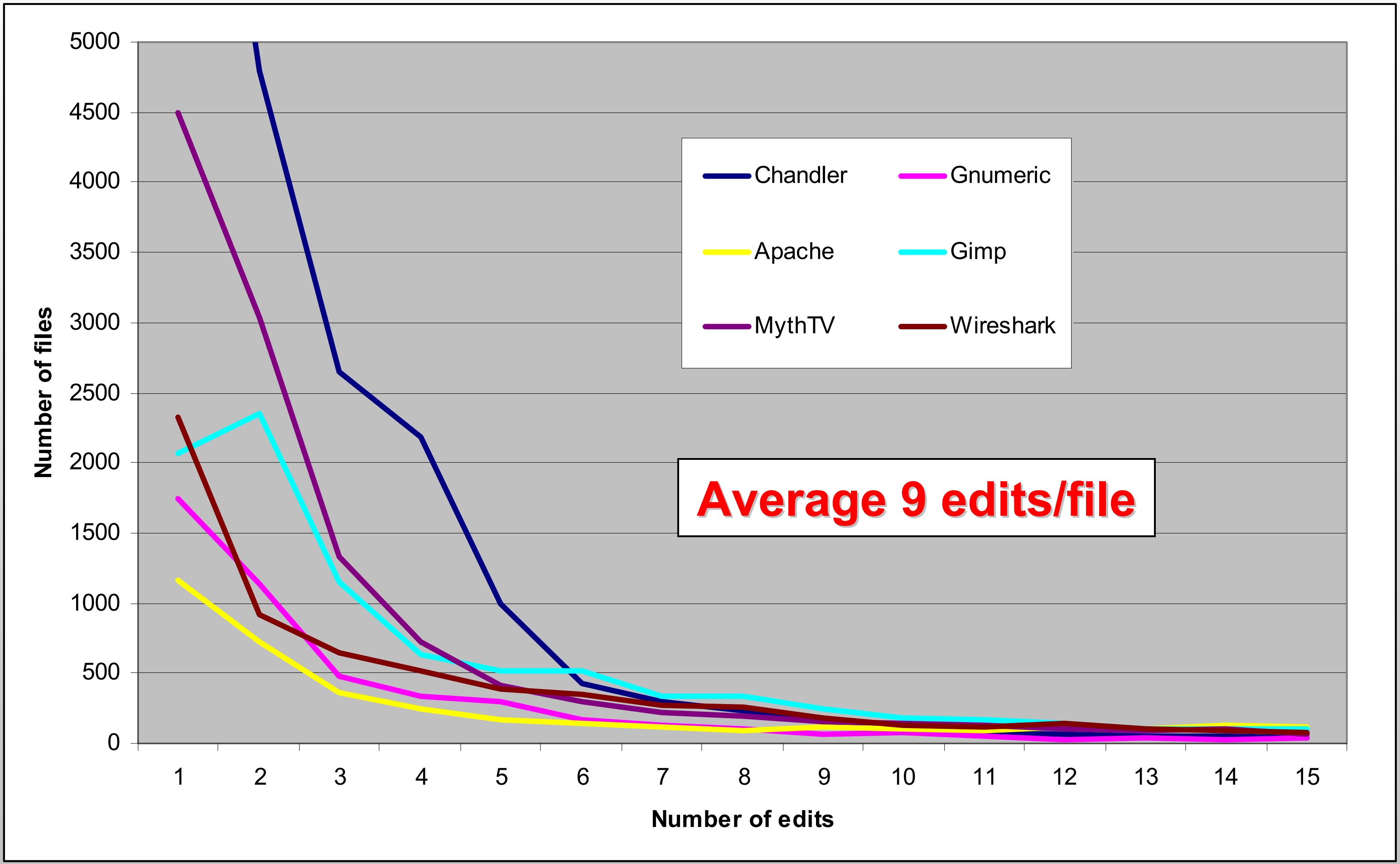
- Demographics of projects
- Analysis of **individuals**
  - Footprint – what parts of the product are focused
  - Change over time
- Analysis of **team**
  - Communication network
  - Visibility overlap
  - System hand-offs
- Analysis of **product**
  - Function call graph
  - Bug duration, origin, severity
- Analysis of **product, team**
  - Overlap in communication and objects

Some results

- Scale of review and rework
  - How many times a file is edited
- System visibility
  - How much of the product does each one see
- System overlap
  - Who worked on whose files
- Evolution of personal footprint

Scale of review and rework

Histogram of number of edits on a file



Average 9 edits/file

System visibility

How much do they work on?

Top ten developers and average on each project

Chandler	Apache	Gimp	Gnumeric	MythTV	Wireshark	Songbird
36%	67%	61%	71%	35%	44%	39%
32%	35%	57%	29%	15%	36%	25%
29%	22%	31%	14%	15%	34%	23%
24%	21%	18%	13%	14%	32%	22%
14%	21%	8%	12%	13%	25%	20%
12%	20%	8%	10%	12%	24%	13%
5%	14%	7%	9%	10%	23%	12%
3%	13%	7%	8%	10%	13%	8%
3%	12%	7%	7%	9%	12%	6%
3%	12%	5%	7%	8%	12%	4%
4.0%	4.6%	1.0%	1.2%	4.7%	8.1%	6.5%

% of files in the project that have at least one edit by the member

System overlap

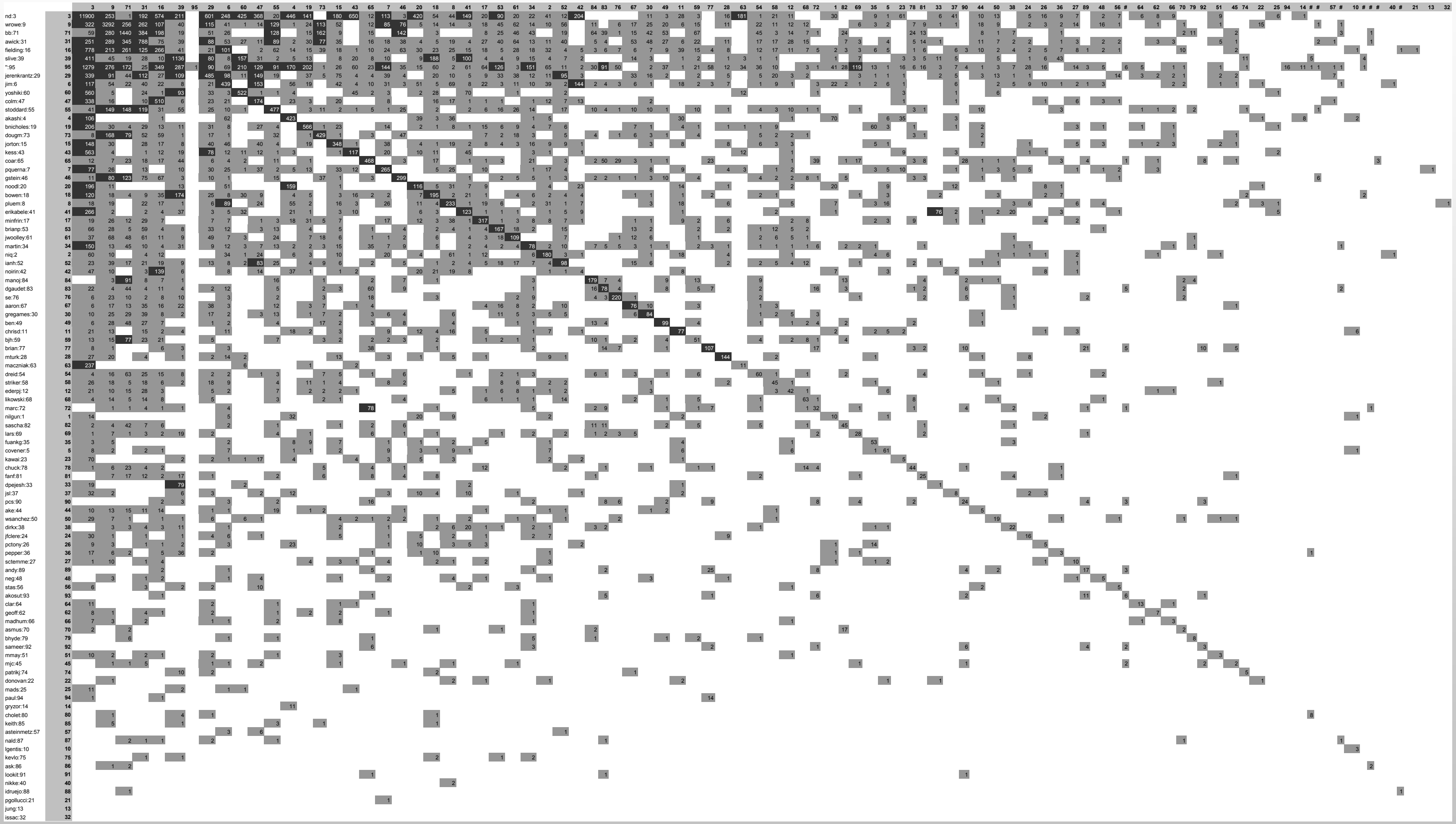
Who worked on whose files (1st order)

	32	21	7	8	12	14	15	35	13	11	3	20	18	5	2	26	23	37	40	4	22	10	6	27	29	9	36	16	34	33	24	39	17	19	41	25	28	30	31	44	38	42	43	1
"": 32	6	3076	1837	3767	1449	1631	670	6061	412	84	87	130	170	204	93	86	79	95	69	405	51	43	36	107	10	7	4	8	4	11	22		7		6	30	10	4	12					
davids: 21		8839	16	1566	24	285	2		5	4	6	10	2		2	5	12			3	4		5		2	1		4		11						1		1						
vajda: 7		19	7686	329	325	340	243	25	96	106	86	46	100	74	111	18	29	50	12	2	62	35	15		2	18	2	6	3			2	3	7	2						2			
bear: 8		2412	230	1359	174	937	44		37	48	33	25	15	1	36	33	1			82	2	38	15	4	2	24		3			3			8										
eikki: 12		2511	297	199	1114	101	139	41	96	31	44	15	49	12	41	13	8	2		2	20	23	7			4	2	5		16	1		2	1										
john: 14		90	304	926	112	1715	219	5	74	167	103	106	48	147	66	108	186	30	17	19	17	26	30	1	7	17	7	11	6			1	2	1	2									
morgen: 15	2	4	309	80	197	224	2561	32	139	102	125	73	151	52	76	85	63	22	12	5	19	8	11	3	12	3	8	4	22	2		2	1	1	1			1		4		1		
markie: 35		309	56	259	3096	7	29	286		1						1					3									17											1			
bkirsch: 13		5	166	94	83	147	130	7	675	43	37	35	44	6	53	29	3			21	5	4	5		2	1		3					1	1										
stearns: 11		6	125	28	43	171	114		38	438	55	56	31	6	112	38	14			1	4	4	34	1	3	3		2	1					1				1						
jeffrey: 3		2	93	35	44	90	127		44	66	595	57	19	3	89	5	4			8	3	4	9		2	8		6						1				1						
alecf: 20		10	60	42	7	143	47		33	63	65	569	42	4	36	19	9				12	2	1	2	11			2										1						
pje: 18		2	94	36	29	34	166		44	39	17	20	495	6	29	21	4				9	1	2	1	5			1			1						1	2						
capps: 5			87	12	26	70	92		6	3	9	11	14	507	2	9	29	49	16			2	1				6		1			19				1								
grant: 2		1	116	20	44	65	81		60	89	99	36	31	2	234	6	2			6	2	5	9		1	6		7					2	2										
donn: 26		3	46	24	21	98	77	1	34	44	6	31	19	7	9	338	45	6			7	5		16	4				4		10						3							
jed: 23		12	20	8	12	199	71		7	20	5	10	3	32	3	32	308	11	33	2	1				5		2		3						3									
pavlov: 37			58	4	6	29	35		4			3	1	49		4	12	292			6																							
andy: 40			28			17	34							20			28	17	255								4			1													1	
mimi: 4		4	5	224	2	17	5			14	4	1		1	6					100								1																
wl: 22		2	71	7	6	23	28	1	11	11	4	11	14	4	1	4	1				112	1	2		1						1													
dan: 10			39	31	34	29	11		4	3	6	2		1	4						1	135	2										5											
ae: 6		2	12	20	12	27	5		3	28	7	9		2	16					1		1	91			5		3																
livier: 27				2	1	1	2		3				4			23								119							8						3	1						
brendano: 29		3	1	2		5	10			4	3	15	7	1	1	4	4								78													1						
bind: 9			17	10	6	20	2			6	5		1	2	1					1			1			39																		
ducky: 36	1			1	1	2	7							9			5	4	1								17					1			1									
pbossut: 16		2	3	2	5	8	4		1	3	6	1	2		1	1				1			1					7																
lisa: 34			4		3	8	10		2				1	1		7	2	2											5															
bin: 33		15		11	1	1								1			1		1											9														
aparna: 24				1	2	5			1			1				6						1		13							7		1											
skinner: 39}			4								4			14			2		1													2												
mikeal: 17			8		1	3									1						5							1			1		6											
mmmm: 19			4	6	2	2	2			1	1			1	1													1						4										
chao: 41			5											7			1													1					11									
{sheila: 25				10																1		1														8								
anthony: 28																3								9													7							
arel: 30		1		1						2		1			2	1								1	4													2						
chaolam: 31						1							7																															
atotic: 44			1				3																																	4				
jbotz: 38						2		1																																4			
michael: 42																																			1									
ys: 43																	1																											
jared: 1																																												

Chandler: 43 developers

System overlap

Who worked on whose files (1st order)



Apache: 94 developers

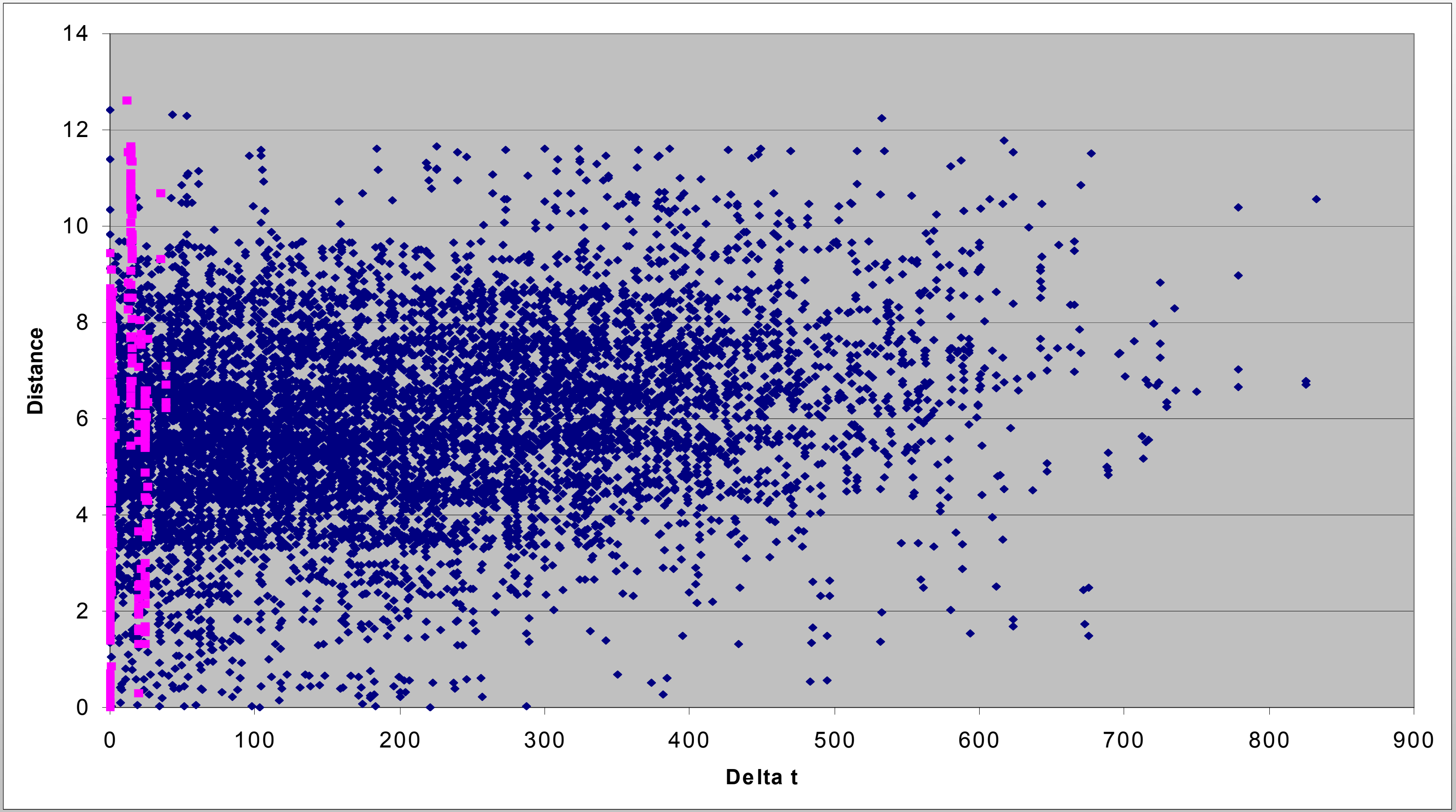
System overlap

Who worked on whose files (entire project)

	eikki	davids	bear	markie	john	vajda	morgen	grant	bkirsch	pje	stearns	jeffrey	capps	alecf	donn	jed	wl	mimi	ae	pbossut	pavlov	dan	bind	andy	brendano	livier	lisa	mmmm	mikeal	bin	ducky	skinner	arel	aparna	jared	chao	jbotz
eikki		2519	349	3106	262	509	306	229	187	166	131	123	76	90	83	55	60	22	52	51	40	54	29	7	15	6	11	9	10	25	3	4	5	8	8	1	3
davids	2519		2998	2314	348	23	21	11	21	10	13	11	8	11	10	27	9	12	16	11	3	3	26	2	7	5		1		24	1	1	4		1	1	1
bear	349	2998		355	1182	494	217	262	236	123	134	108	61	123	95	40	46	280	63	44	15	74	42	6	17	9	2	12	15	12	2	1	6	7	10	1	3
markie	3106	2314	355		31	75	50	13	10	9	8	6	4	5	6	5	11		2	2	3	2	3		1	1	2			31	2	1			3		1
john	262	348	1182	31		454	273	261	238	145	175	132	218	140	118	115	52	78	48	52	54	61	32	51	17	6	16	11	16	5	18	5	4	7	3	7	3
vajda	509	23	494	75	454		369	340	282	228	189	162	152	155	126	62	87	20	52	48	95	94	32	32	19	7	19	13	13	2	6	10	5	7	12	5	3
morgen	306	21	217	50	273	369		196	177	249	128	114	180	109	99	101	72	20	40	40	66	22	16	47	15	10	22	9	6	5	12	8	3	1	2	4	3
grant	229	11	262	13	261	340	196		207	148	152	131	53	123	64	27	36	61	46	54	9	54	25	2	16	6	2	10	13			2	4	2	15	1	2
bkirsch	187	21	236	10	238	282	177	207		145	124	109	58	115	73	34	44	44	34	55	17	30	22	4	15	8	5	6	7		1	3	4	4	10	1	1
pje	166	10	123	9	145	228	249	148	145		115	75	67	106	80	31	57	11	21	23	25	10	12	2	13	10	16	3	2			4	3	1	1		3
stearns	131	13	134	8	175	189	128	152	124	115		95	55	99	78	37	43	38	46	34	13	24	22	4	16	7	11	7	6			4	4	2	1	1	3
jeffrey	123	11	108	6	132	162	114	131	109	75	95		45	70	43	26	25	21	35	40	9	31	19	2	12	6	6	5	8			8	4	5	5	1	1
capps	76	8	61	4	218	152	180	53	58	67	55	45		48	46	99	20	8	17	18	76	9	13	55	15	5	20	6	3	4	24	13	3	1		9	
alecf	90	11	123	5	140	155	109	123	115	106	99	70	48		56	34	42	12	32	24	17	11	19	1	17	7	4	5	3			2	5	1		1	
donn	83	10	95	6	118	126	99	64	73	80	78	43	46	56		46	30	8	22	15	19	20	11	3	13	21	13	4	3	1		5	4	18		1	
jed	55	27	40	5	115	62	101	27	34	31	37	26	99	34	46		11	10	19	13	23	4	14	36	14	5	7	4	1	4	12	5	3		1	2	1
wl	60	9	46	11	52	87	72	36	44	57	43	25	20	42	30	11		3	13	8	17	7	5	2	6	5	6	4	2		1	2	2	1	1	1	1
mimi	22	12	280		78	20	20	61	44	11	38	21	8	12	8	10	3		8	23	1	1	7	1	3	1		2				1	2			1	
ae	52	16	63	2	48	52	40	46	34	21	46	35	17	32	22	19	13	8		14	4	7	18	1	10	4		3	2			1	4	1	1	1	1
pbossut	51	11	44	2	52	48	40	54	55	23	34	40	18	24	15	13	8	23	14		2	9	12	2	5	4		8	2			1	2			1	
pavlov	40	3	15	3	54	95	66	9	17	25	13	9	76	17	19	23	17	1	4	2		2	3	43	4	3	8	1			7	5	1			2	
dan	54	3	74	2	61	94	22	54	30	10	24	31	9	11	20	4	7	1	7	9	2		5		3	3		1	13				1	3			
bind	29	26	42	3	32	32	16	25	22	12	22	19	13	19	11	14	5	7	18	12	3	5		1	5	3	1	3				1	4		1	1	1
andy	7	2	6		51	32	47	2	4	2	4	2	55	1	3	36	2	1	1	2	43		1		1	1		2		3	9	2	1			1	2
brendano	15	7	17	1	17	19	15	16	15	13	16	12	15	17	13	14	6	3	10	5	4	3	5	1		3	1	3	1			1	6		1	1	
livier	6	5	9	1	6	7	10	6	8	10	7	6	5	7	21	5	5	1	4	4	3	3	3	1	3			2	1			1	1	17		1	
lisa	11		2	2	16	19	22	2	5	16	11	6	20	4	13	7	6				8		1		1			1				5					
mmmm	9	1	12		11	13	9	10	6	3	7	5	6	5	4	4	4	2	3	8	1	1	3	2	3	2	1		1	1		1	1			1	
mikeal	10		15		16	13	6	13	7	2	6	8	3	3	3	1	2		2	2		13			1	1		1						1			
bin	25	24	12	31	5	2	5						4		1	4								3				1			1					1	
ducky	3	1	2	2	18	6	12		1				24			12	1				7			9						1		1				3	
skinner	4	1	1	1	5	10	8	2	3	4	4	8	13	2	5	5	2	1	1	1	5		1	2	1	1	5	1			1		1			1	
arel	5	4	6		4	5	3	4	4	3	4	4	3	5	4	3	2	2	4	2	1	1	4	1	6	1		1				1				1	
aparna	8		7		7	7	1	2	4	1	2	5	1	1	18		1		1			3				17			1								
jared	8	1	10	3	3	12	2	15	10	1	1	5				1	1		1				1		1												1
chao	1	1	1		7	5	4	1	1		1	1	9	1	1	2	1	1	1	1	2		1	1	1	1		1		1	3	1	1				
jbotz	3	1	3	1	3	3	3	2	1	3	3	1				1	1		1				1	2											1		
sheila		1	10		4	1			1				1					2				1															
anthony			2				3		1	3					3											9											
chaolam			4		3		1		1	7	1				3																						
''	1		1	1			3	1																							1						
atotic	1			1		2	3																														
ys							1						1			1								1													
michael					1		1																													1	

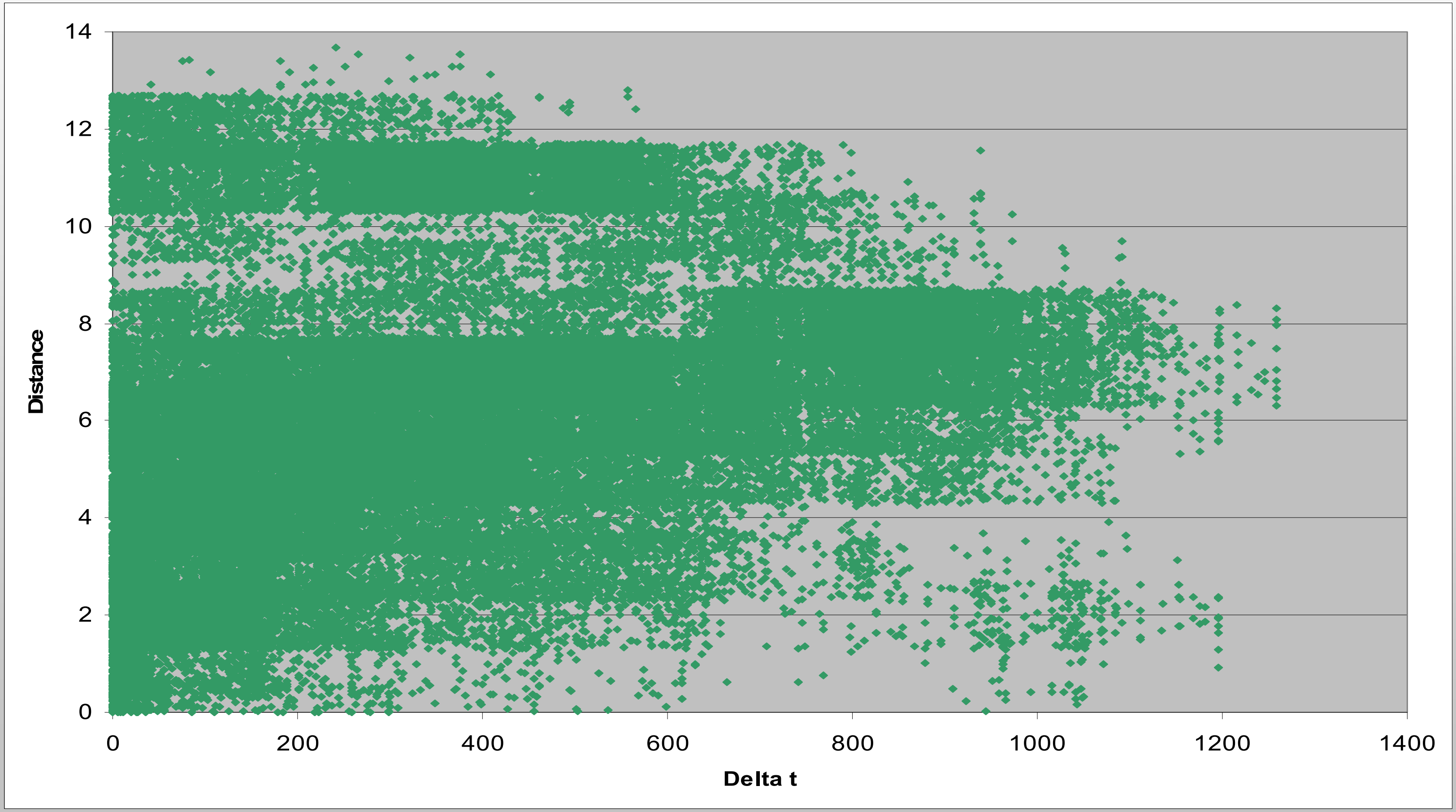
Evolution of personal footprint

Δt v Distance



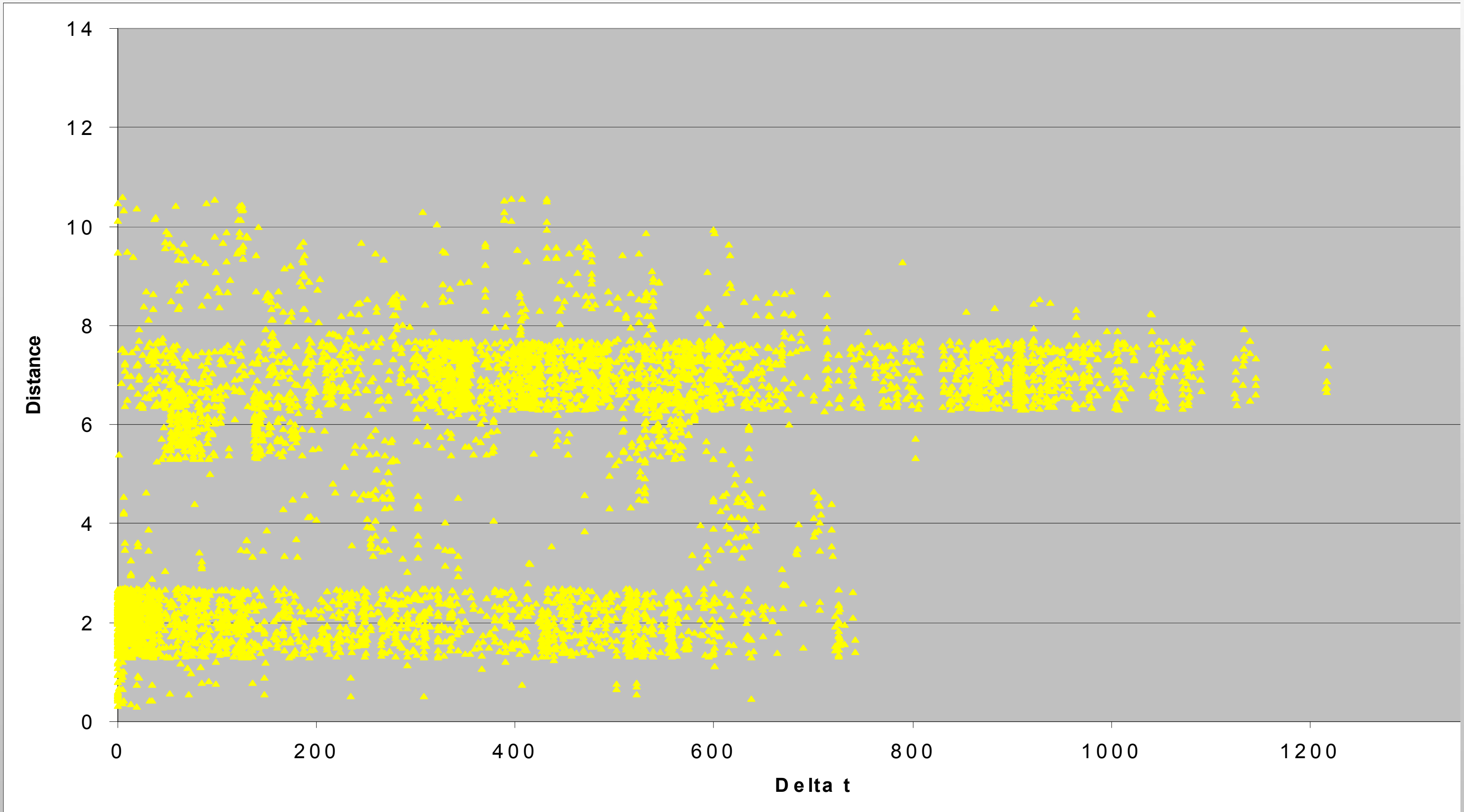
Evolution of personal footprint

Δt v Distance



Evolution of personal footprint

Δt v Distance



Next steps

- Integrate in analysis the ***bug*** and ***mailing list*** data
- Analyze relationship between
  - Product and communication
  - Analyze problem solving over time and product structure
- Analyze using FCG instead of folder hierarchy for product structure

Thank you