Organizational Volatility and its Effects on Software Defects

Audris Mockus

audris@avaya.com

Avaya Labs Research

Basking Ridge, NJ 07920

http://mockus.org/

Outline

Motivation

Related Work - Organization and Defects

Data Archeology and Measures

Model setup

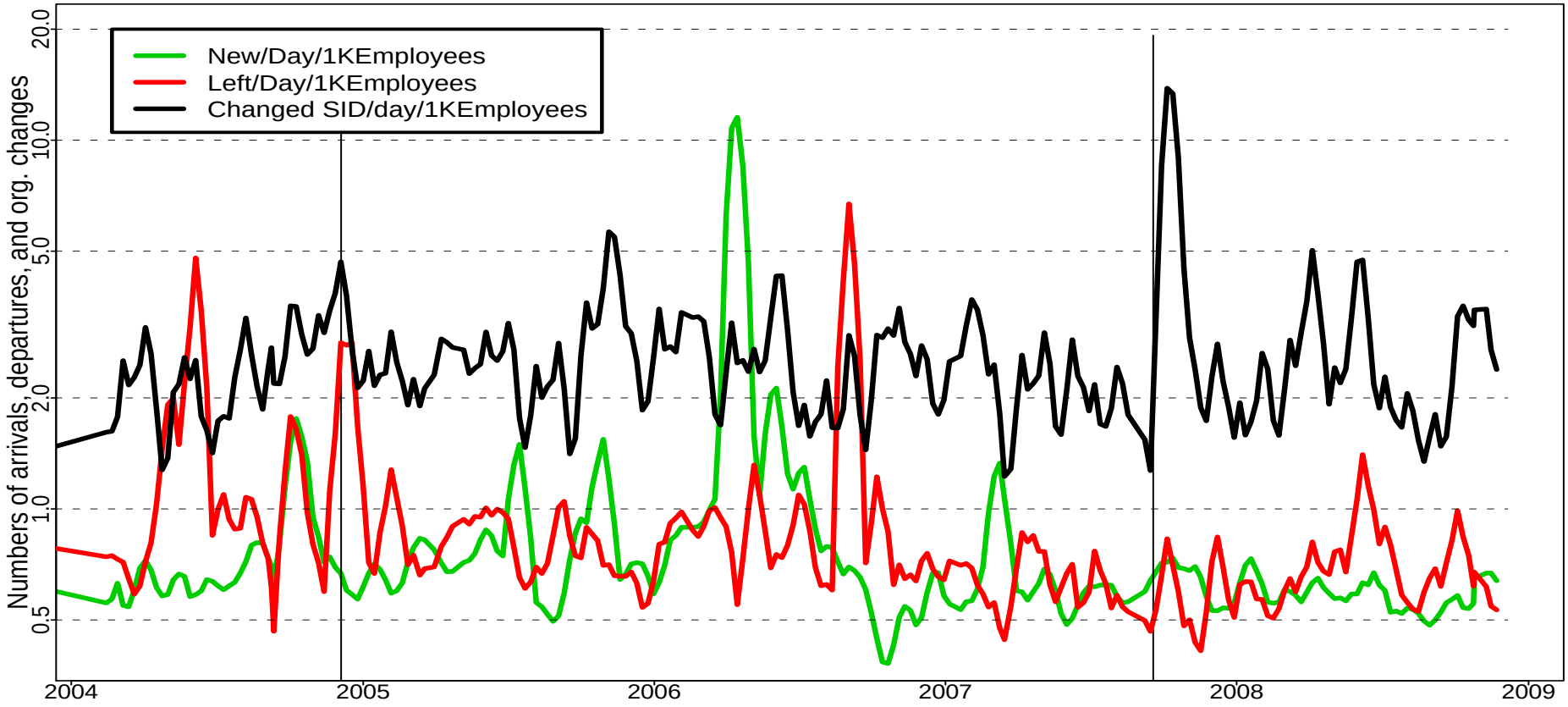
Results

Discussion

Motivation

- Theory
  - The key premise of Organization:
    - Produce things an individual can't produce
    - Produce more efficiently Taylor [1911]
  - Organizational design: reshaping organization's structure (i.e., formal reporting relationships) to improve the organization
- Reality
  - Organization's structure is often changing
  - How are these changes reflected in the organization's efficiency?
    - Developer productivity
    - **Product quality**

Staff volatility: 3-week averages



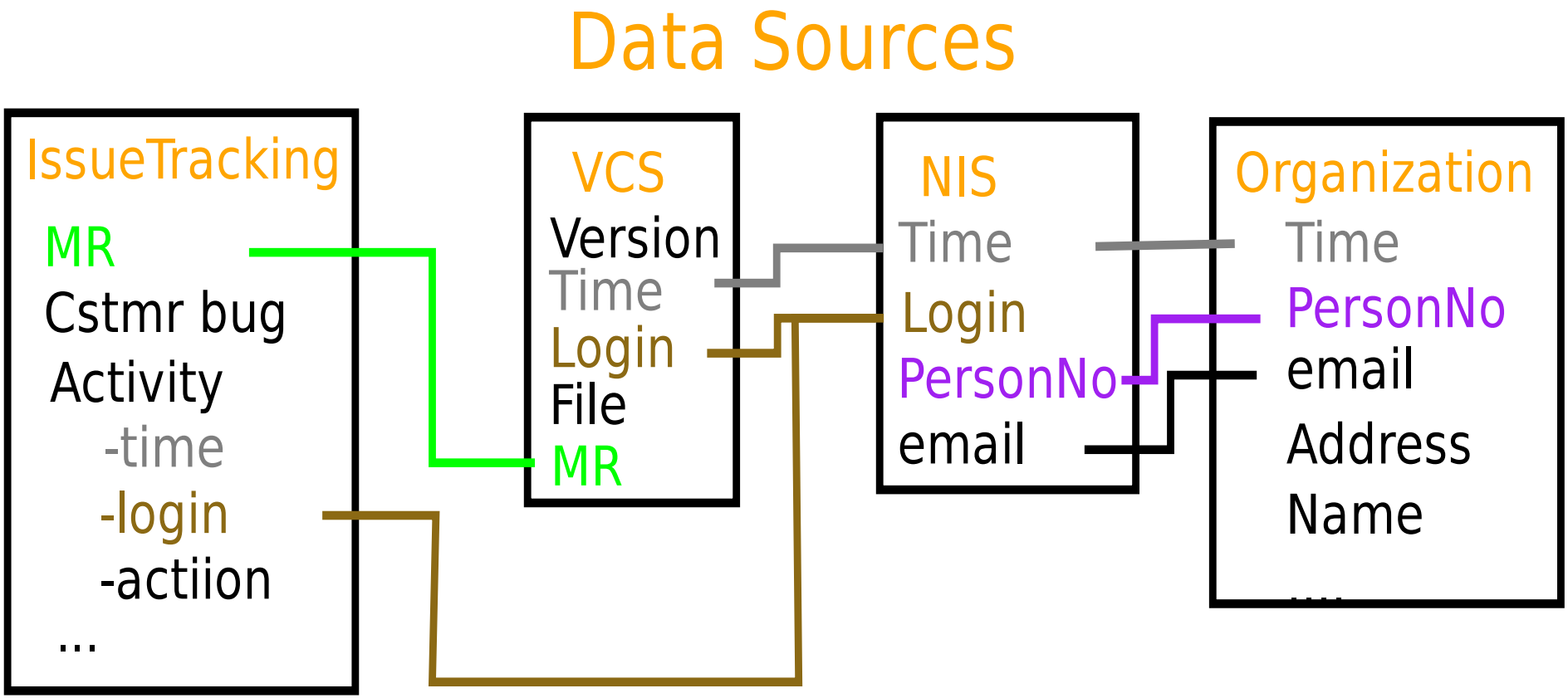
An avg. person: 1.4 **years** btwn **org changes** (**two yrs.** for product)

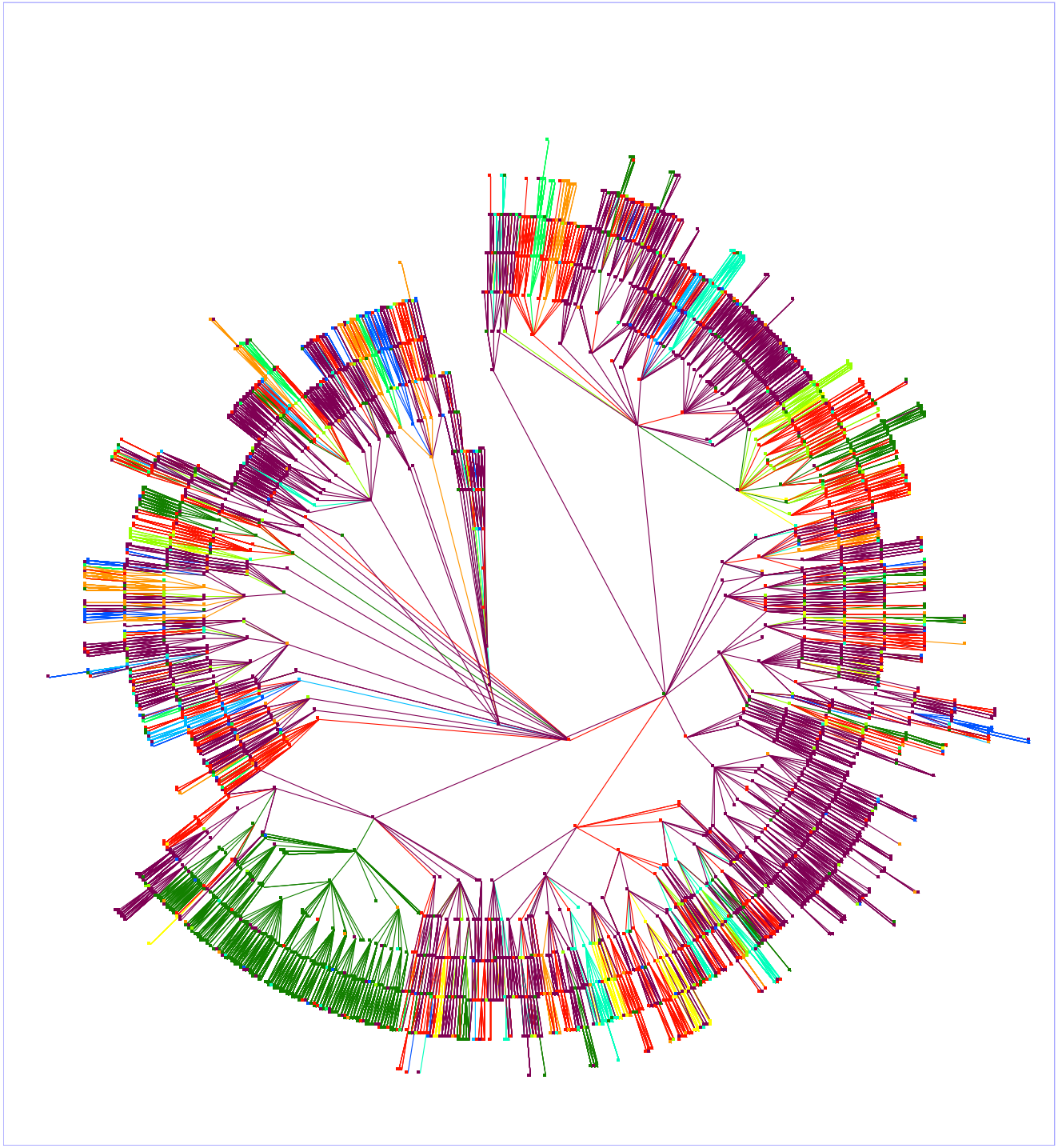
A 20-person organization: 5 **leave/join** per year on average

Archival Data, Social Capital, Survivors

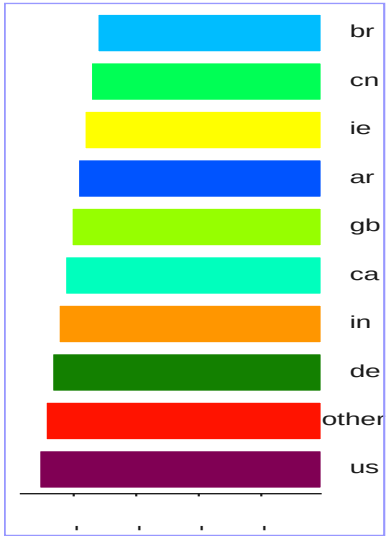
- Method: **"the palest ink is clearer that the best memory"** Webb et al. [1966], Geisler [1999] propose to measure organizational change based on archival records.
- **What lowers turnover?** *Social Capital* Cohen and Prusak [2001]: effectiveness through culture of trust and respect, generous benefits, and recognition of importance of peoples personal lives.
- **Studies of survivors** Armstrong-Stassen [2005]
  - Increased workload demands
  - Increased usage of escape coping strategies
    - Control-based coping: positive thinking, direct-action, and instrumental support
    - Escape coping: avoidance and disengagement
  - Higher incidence of health problem symptoms

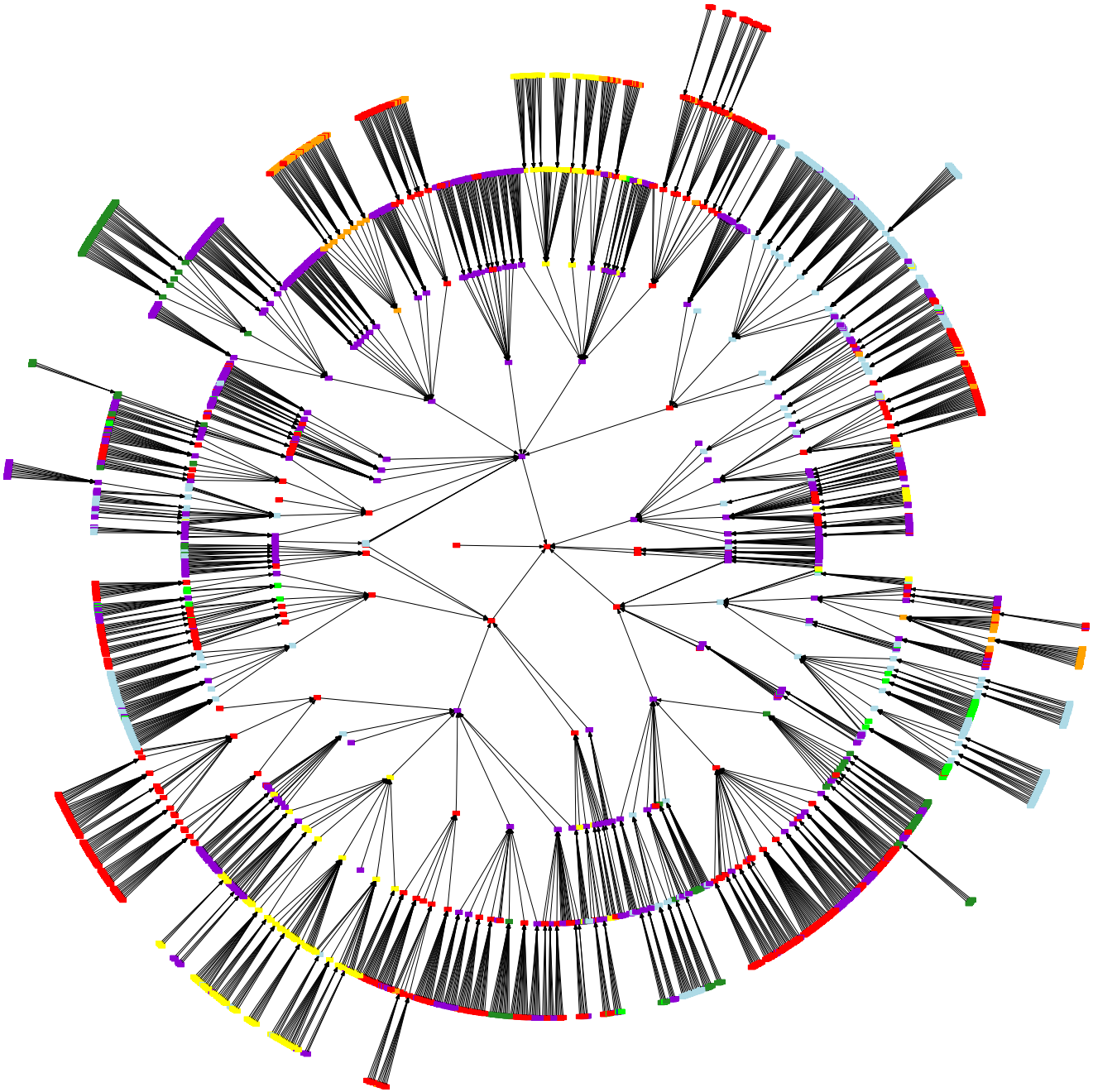
Digital Archeology: Sources





Organizational Hierarchy Colored by Location



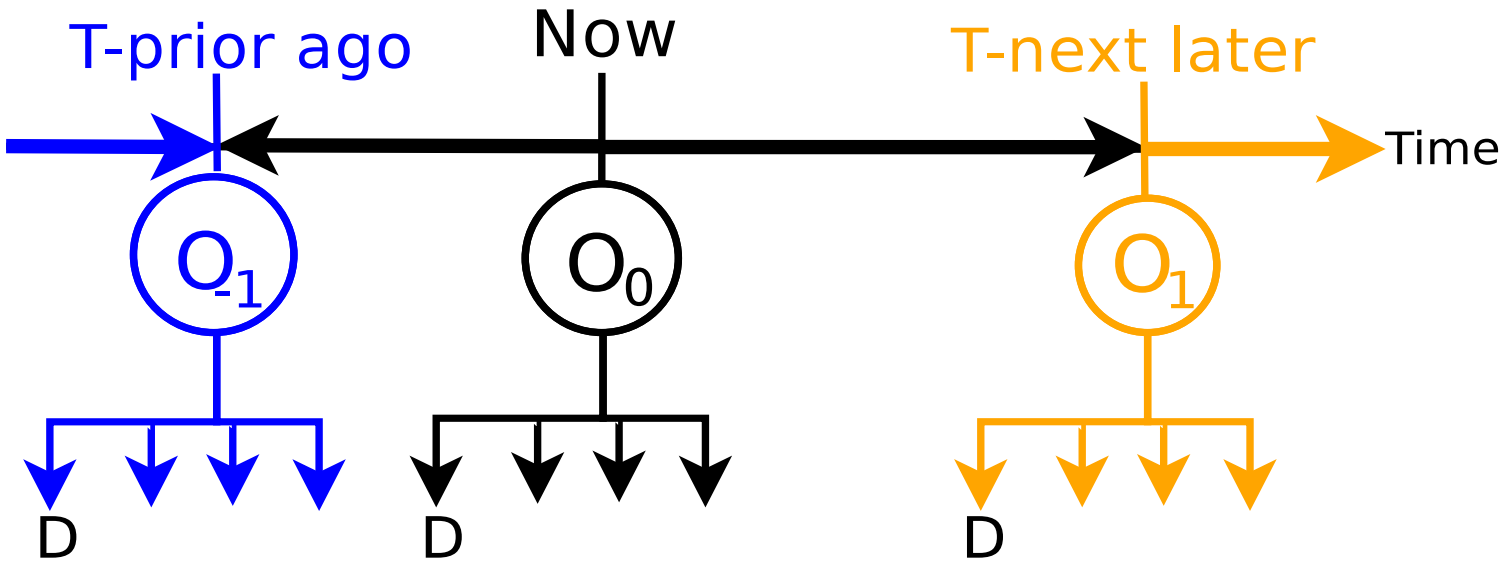


Product Development Hierarchy Colored by Location

Operationalizations of volatility

Concept	Operationalization
Proximity in time to the organizational change	Time (in years) until the next and after the last change in the organization ID
Size of the reorganization	Number of employees leaving the organization over past two months
New recruits	Number of employees entering the organization over past two months
Size of the organization	Number of employees within the organization
Other factors	Product, Location, Organization ID, Developer ID

Proximity to the Organizational Change



- Now D works in O_0

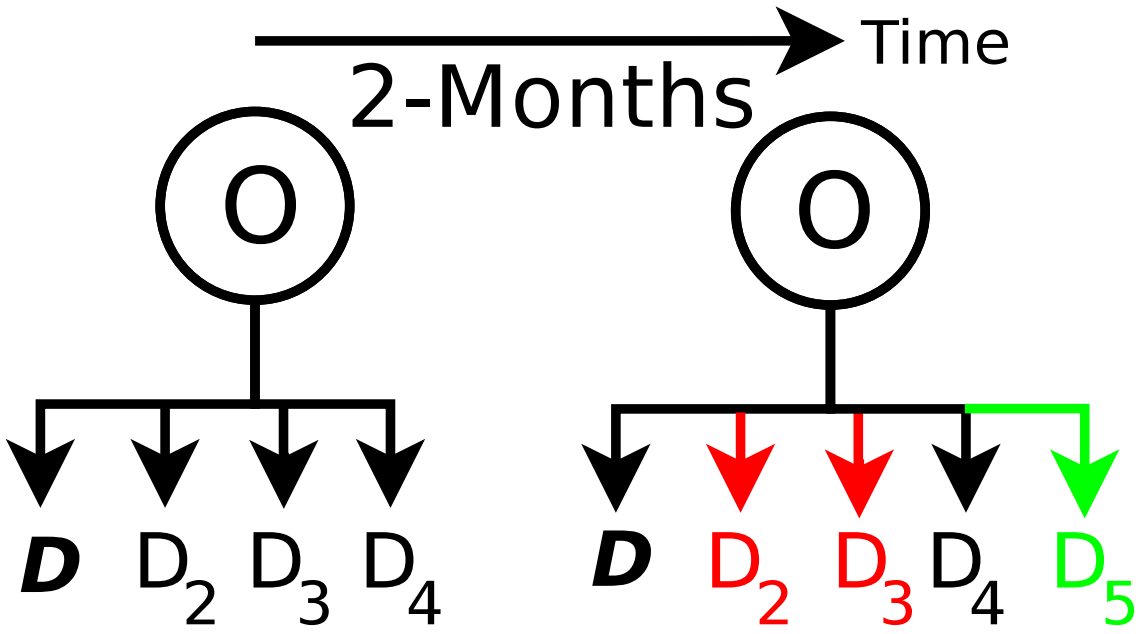
- T-prior ago D's org changed from O_{-1} to O_0

- In T-next time D's org will change to O_1

$$T_{Next}(l,t) = \arg\min_{s>0, O(l,t+s)\neq O(l,t)} O(l,t+s)$$

$$T_{Prior}(l,t) = \arg\min_{s>0, O(l,t-s)\neq O(l,t)} O(l,t-s)$$

Inflow and Outflow of Colleagues



- For Developer D:

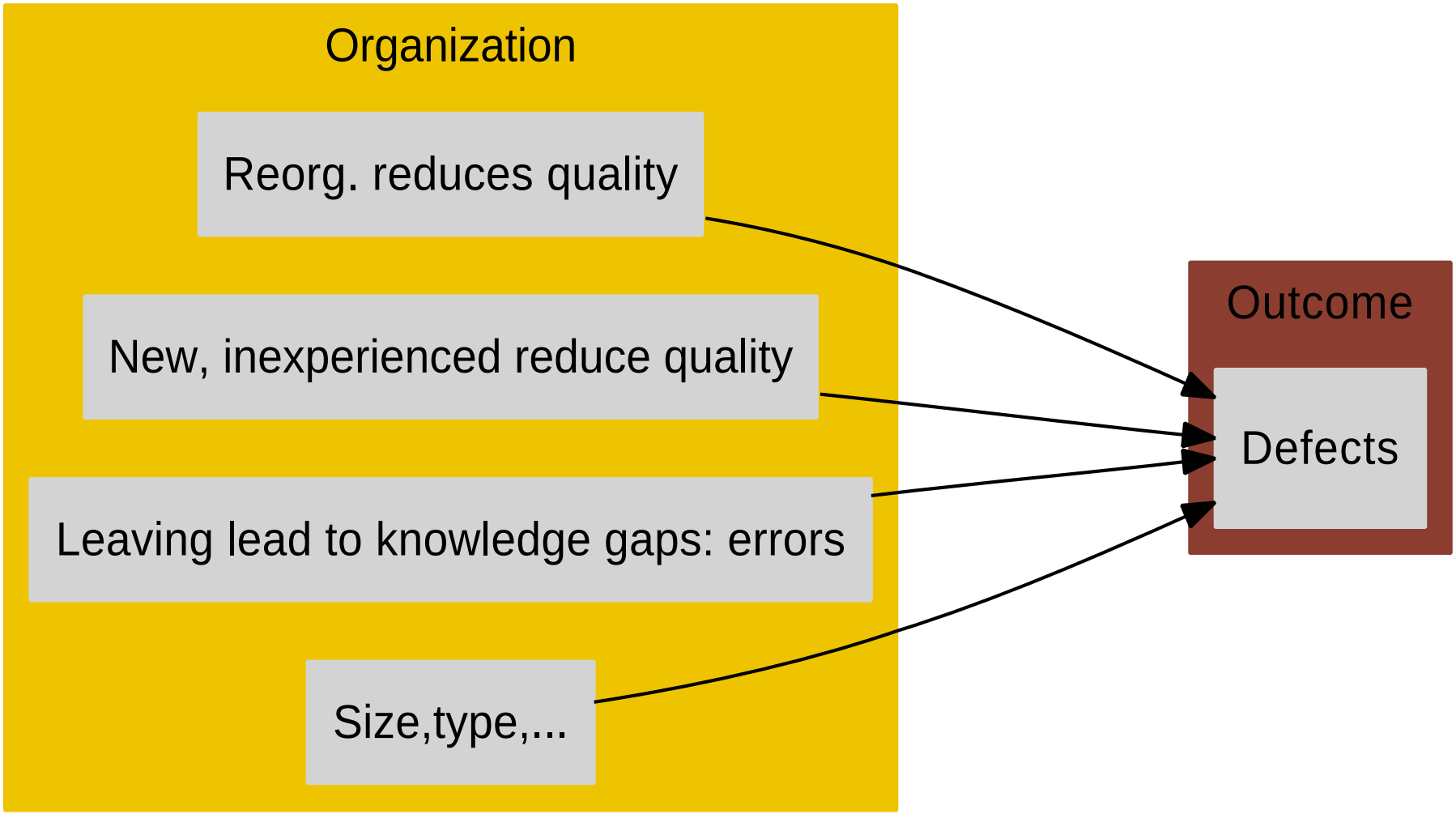
two colleagues left - D_2 and D_3

one colleague arrived - D_5

$$Left(D,t) = \aleph\{d : O(d,t-\delta) = O(D,t) \wedge O(d,t) \neq O(D,t)\}$$

$$New(D,t) = \aleph\{d : O(d,t-\delta) \neq O(D,t) \wedge O(d,t) = O(D,t)\}.$$

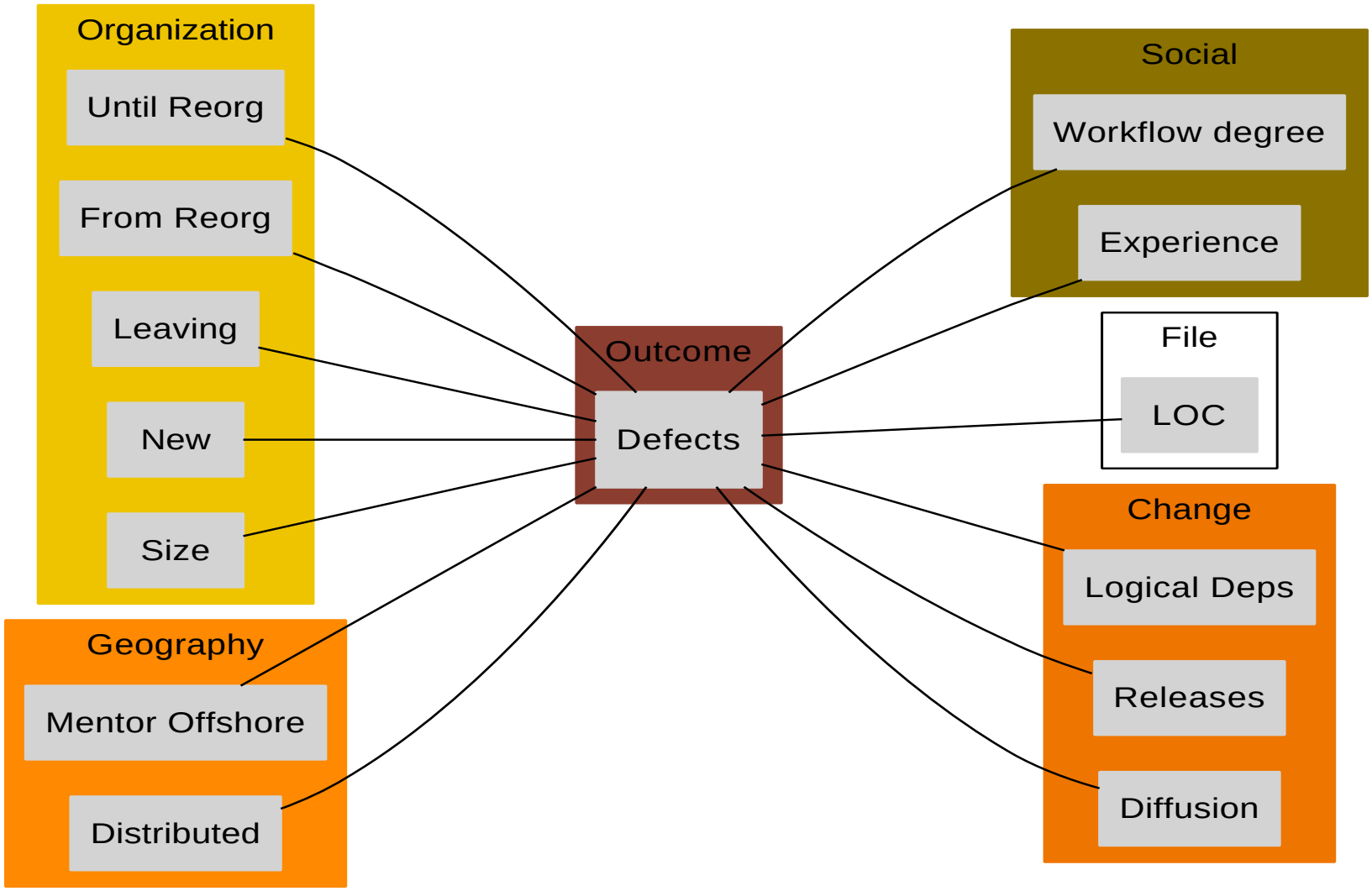
Hypotheses


Organization
Reorg. reduces quality
New, inexperienced reduce quality
Leaving lead to knowledge gaps: errors
Size,type,...
Outcome
Defects

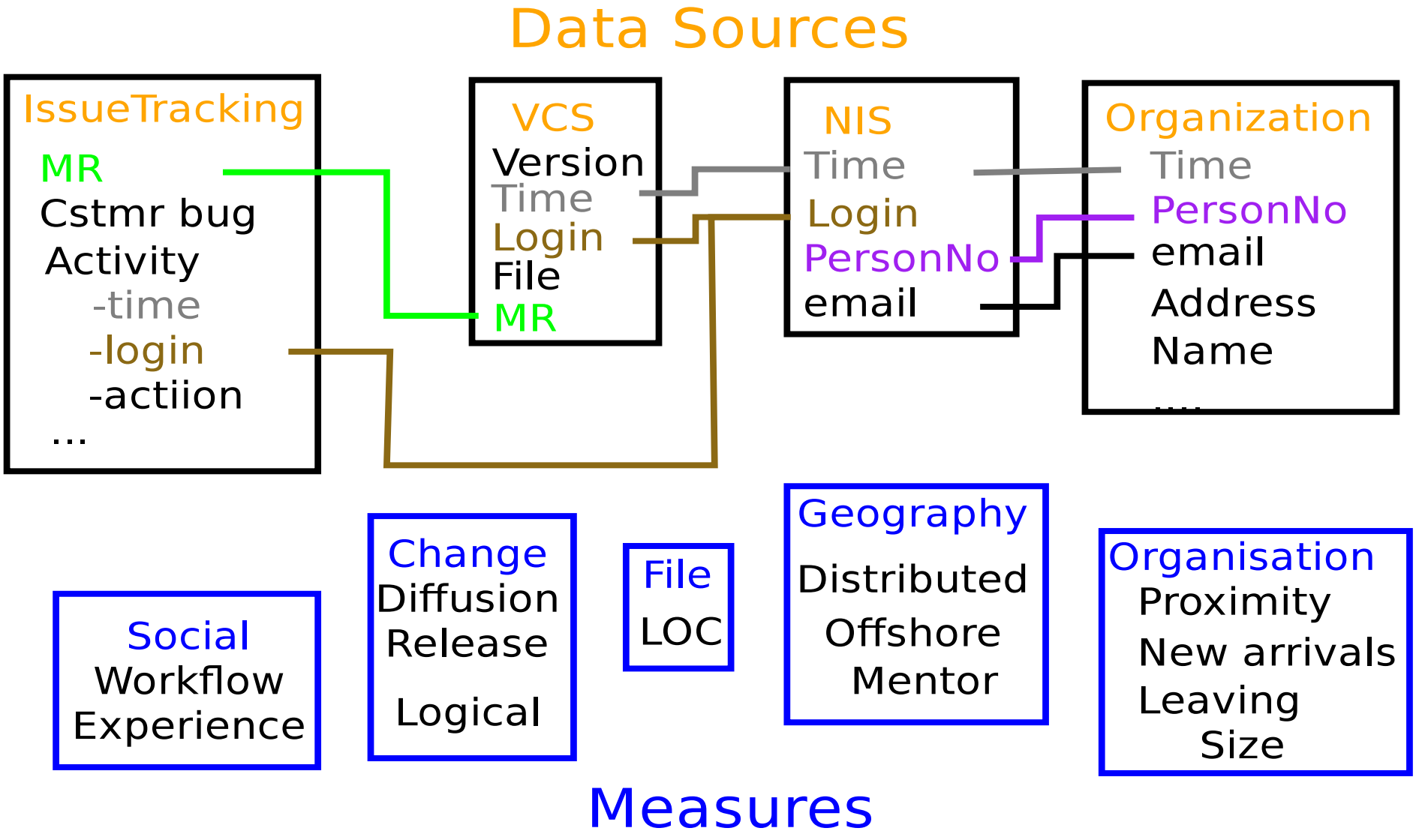
Defect Modeling Objective

- Scientific
  - Does organizational volatility affect defects **after** adjusting for factors known from literature?
  - **Review** related work $\Longrightarrow$ **reproduce** earlier results
- Practical
  - what are relative contributions of predictors?
- Methodological
  - Use history to explain future defects
  - Avoid release- or period-specific anomalies

Related work: replication hypotheses



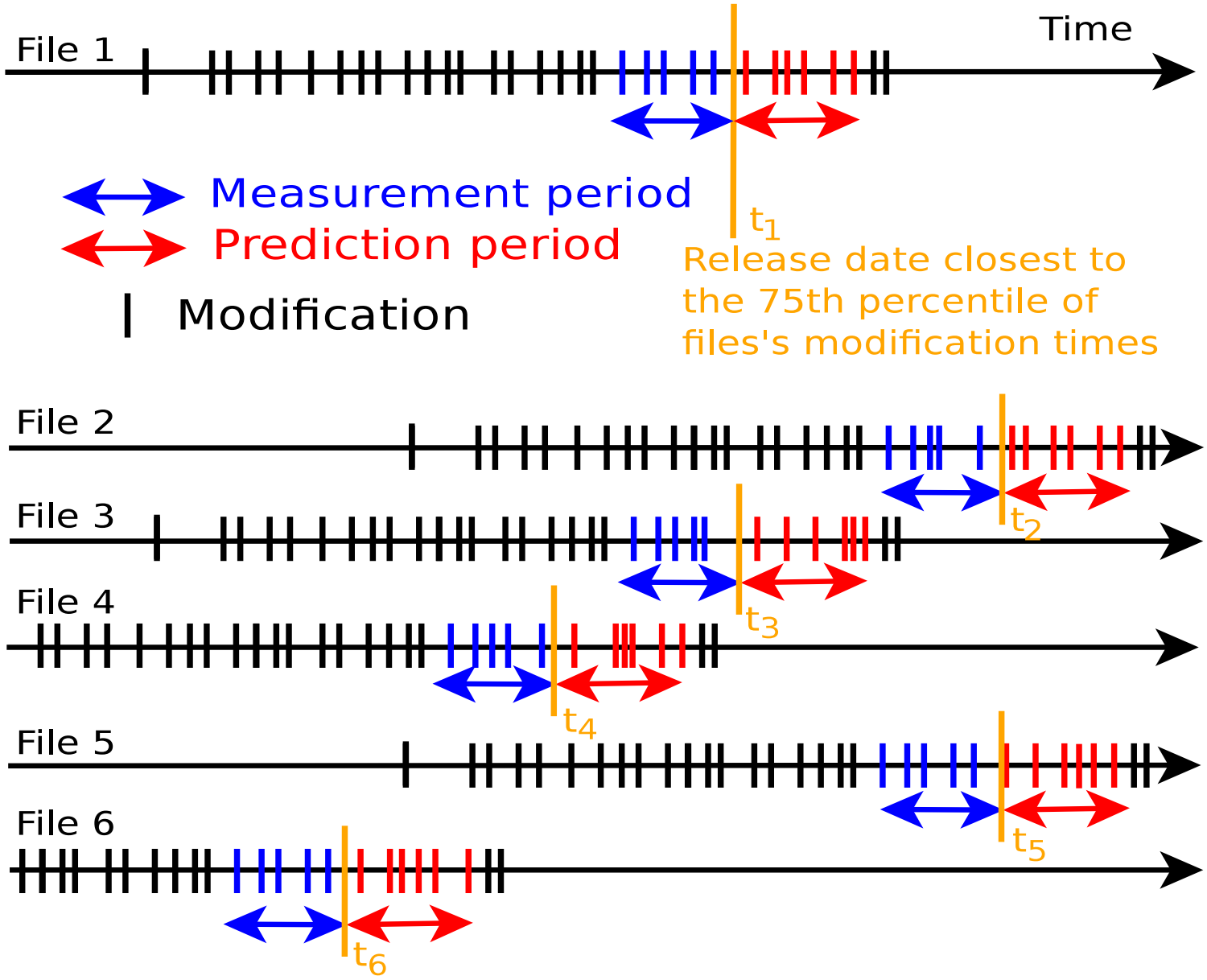
Data Sources and Measures



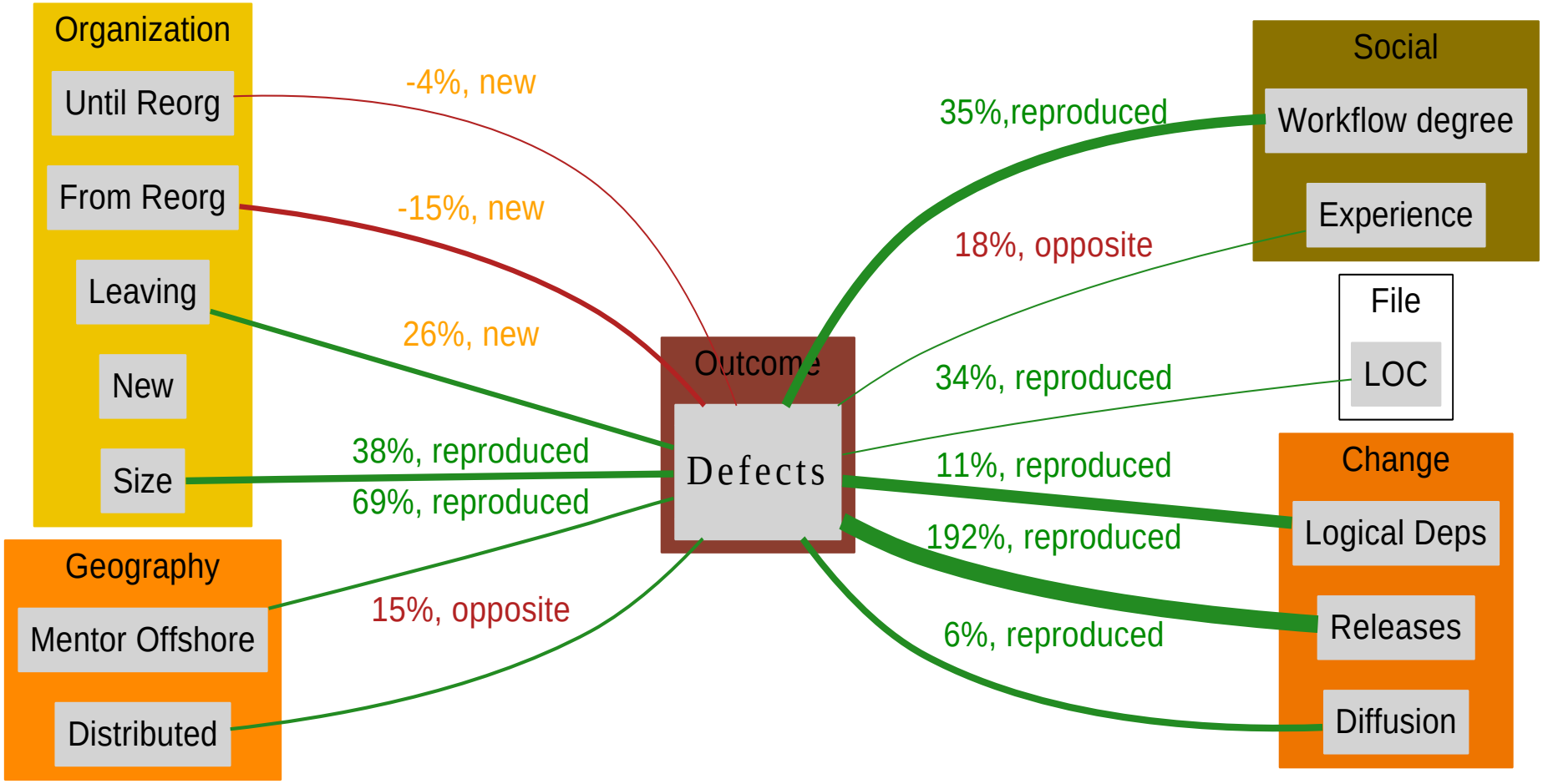
Logistic regression

- File is the observation unit
  - One-year prior to t_f *observation period* to obtain predictors.
  - One-year *prediction period* after t_f to count customer reported defects.
  - t_f is file-specific to avoid peculiarities of a release
  - Organizational measures for a file are derived from developers modifying the file during the *observation period*
  - Outcome: customer reported defect during *prediction period*
- 32099 files, 7% with customer defects

Measurement and *Prediction* Periods



Reproducing earlier results



Conclusions

- The scientific perspective
  - propose and relate to defects three measures of organizational change
  - reproduce results from prior empirical studies.
- The methodological perspective
  - reduce the bias of irrelevant context by modeling different parts of the system at different times
  - evaluate the impact of novel factors after adjusting for factors known to be related to software quality.
- The practical perspective: prioritize quality improvement
  - the magnitude the impact of the organizational change on the probability of customer-reported defects
  - the relative importance of organization-, workflow-, and code-derived factors

References

M Armstrong-Stassen. Coping with downsizing: A comparison of executive-level and middle managers. *International Journal of Stress Management*, 12(2):117–141, 2005.

Christian Bird, Nachiappan Nagappan, Premkumar Devanbu, Harald Gall, and Brendan Murphy. Putting it all together: Using socio-technical networks to predict failures. In *17th International Symposium on Software Reliability Engineering (ISSRE 09)*, Bengaluru-Mysuru, India, 2009a.

Cristian Bird, Nachiappan Nagappan, Premkumar Devanbu, Harald Gall, and Brendan Murphy. Does distributed development affect software quality? an empirical case study of windows vista? In *ICSE 2009*, 2009b.

Marcelo Cataldo, Audris Mockus, Jeffrey A. Roberts, and James D. Herbsleb. Software dependencies, the structure of work dependencies and their impact on failures. *IEEE Transactions on Software Engineering*, 2009. URL `papers/conway.pdf`.

D. J. Cohen and L Prusak. *In Good Company: How Social Capital Makes Organizations Work*. Harward Business Press, 2001.

E Geisler. Organizational change phenomena, managerial cognition, and archival measures: Reconceptualization and new empirical evidence. Technical Report 99-02, Stuart School of Business, Illinois Institute of Technology, August 1999.

J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally-distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–494, June 2003a. URL `papers/delay.pdf`.

James Herbsleb and Audris Mockus. Formulation and preliminary test of an empirical theory of coordination in software engineering. In *2003 International Conference on Foundations of Software Engineering*, Helsinki, Finland, October 2003b. ACM Press. URL `papers/conway.pdf`.

Audris Mockus and David M. Weiss. Predicting risk of software changes. *Bell Labs Technical Journal*, 5(2):169–180, April–June 2000. URL `papers/bltj13.pdf`.

Nachiappan Nagappan, Brendan Murphy, and Victor R. Basili. The influence of organizational structure on software quality: an empirical case study. In *ICSE 2008*, pages 521–530, 2008.

Frederick Winslow Taylor. *The Principles of Scientific Management*. Harper & Brothers, 1911.

E. Webb, D. Campbel, R. Schwartz, and L. Sechrest. *Unobtrusive Measures: Nonreactive Research in the Social Sciences*. Rand McNally Colledge Publishing Company, Chicago, IL, 1966.

Abstract

The key premise of an organization is to allow more efficient production, including production of high quality software. To achieve that, an organization defines roles and reporting relationships. Therefore, changes in organization's structure are likely to affect product's quality. We propose and investigate a relationship between developer-centric measures of organizational change and the probability of customer-reported defects in the context of a large software project. We find that the proximity to an organizational change is significantly associated with reductions in software quality. We also replicate results of several prior studies of software quality supporting findings that code, change, and developer characteristics affect fault-proneness. In contrast to prior studies we find that distributed development decreases quality. Furthermore, recent departures from an organization were associated with increased probability of customer-reported defects, thus demonstrating that in the observed context the organizational change reduces product quality.

Audris Mockus
Avaya Labs Research
233 Mt. Airy Road
Basking Ridge, NJ 07920
ph: +1 908 696 5608, fax:+1 908 696 5402
http://mockus.org, mailto:audris@mockus.org

Audris Mockus is interested in quantifying, modeling, and improving software development. He designs data mining methods to summarize and augment software change data, interactive visualization techniques to inspect, present, and control the development process, and statistical models and optimization techniques to understand the relationships among people, organizations, and characteristics of a software product. Audris Mockus received B.S. and M.S. in Applied Mathematics from Moscow Institute of Physics and Technology in 1988. In 1991 he received M.S. and in 1994 he received Ph.D. in Statistics from Carnegie Mellon University. He works in Avaya Labs Research. Previously he worked in the Software Production Research Department of Bell Labs.

Reproducing earlier results

Class	Predictor	Effect	Propstns	Reproduced
Org. chng	Size of org.	38%	control	+Nagappan et al. [2008]
	From prior (yrs)	-15%	+1	new result
	Until next (yrs)	-4%	+1	new result
	Left	26%	+4	new result
	Newcomers	N/A	−3,2	new result
File	LOC	34%	+5	+various
	Logical Deps.	11%	+6	+Cataldo et al. [2009], Bird et al. [2009a]
Change	Release Deps.	192%	+10	+Herbsleb and Mockus [2003b]
	Change Diffusion	6%	+6	+Mockus and Weiss [2000]
Social	Workflow Deps.	35%	+7	+Cataldo et al. [2009], Bird et al. [2009a], Herbsleb and Mockus [2003b]
	Experience (yrs)	18%	+8	−Mockus and Weiss [2000]
Geo.	Distributed	15%	+9	−Bird et al. [2009b],+Herbsleb and Mockus [2003a]
	Mentor offshore	69%	+9	new result

32099 files, 7% with customer defects, 41% of deviance explained

Class	Predictor	Est.	StdErr	p-val	Devnc
File	log(LOC)	0.43	0.03	0.00	2450
Chng	log(Logical)	0.25	0.02	0.00	978
	log(Releases)	2.67	0.07	0.00	2331
	log(Diffusion)	0.08	0.03	0.00	321
Socl	log(Workflow)	0.43	0.05	0.00	255
	log(Experience)	0.28	0.04	0.00	13
Geo	Distributed	0.14	0.07	0.04	41.94
	Mentor	0.53	0.12	0.00	27.97
Org	log(OrgSize)	0.48	0.06	0.00	160
	log(From)	−0.40	0.07	0.00	51
	log(Until)	−0.06	0.03	0.09	6
	log(Left + 1)	0.33	0.04	0.00	74
	log(New + 1)	−0.01	0.04	0.70	0

Class	Predictor	Description
Organization	Size of organization	$\max_{l:(l,f,t-1\leq t_o\leq t)} S(l,t_o)$
	Time from prior change	$\min_{l:(l,f,t-1\leq t_o\leq t)} P_{prior}(l,t_o)$
	Time until next change	$\min_{l:(l,f,t-1\leq t_o\leq t)} P_{next}(l,t_o)$
	Number leaving org.	$\max_{l:(l,f,t-1\leq t_o\leq t)} L(l,t)$
	Number of newcomers	$\max_{l:(l,f,t-1\leq t_o\leq t)} N(l,t)$
File	LOC	Lines of non-commentary source code
Change	Logical Deps.	The number of other files changed by the past MRs modifying the file: $LD(f,t) = \aleph\{f_o : \exists mr, \exists t_1, t_2 \leq t,\ (f_o, mr, t_1) \wedge (f, mr, t_2)\}$
	Release Deps.	The maximum number of releases an MR is submitted to over MRs modifying the file during the *measurement period*: $R(f,t) = \max_{mr} \aleph\{r : \exists t_o \leq t,\ (r, mr, t_o)\}$
	Change Diffusion	The maximum number of files changed by an MR modifying the file during the *measurement period*: $D(f,t) = \max_{mr} \aleph\{f_o : \exists t_o \leq t,\ (f_o, mr, t_o)\}$
Social	Workflow	The maximum degree of the workflow network over developers modifying the file during the *measurement period*: $W(f,t) = \max_l \aleph\{l_o : \exists t_1 \leq t, \exists t_2 \in [t-1,t],\ (l,f,t_2) \wedge (l_o,l,t_1)\}$
	Years of prj. experience	The minimum of the years of experience over all developers modifying the file during the *measurement period*.
Geography	Distributed development	The number of sites that modified the file during the *measurement period*
	Mentor Offshore	The maximum of the indicator that a mentor is in another site over developers modifying the file during the *measurement period*

ANOVA and regression of developer productivity ($R^2 = 0.46$)

Predictor	Df	Sum Sq	Mean Sq	F value	Pr(>F)	C
HRID	1226	33918.00	27.67	17.10	0.00	
SID	267	1752.13	6.56	4.05	0.00	
log(Newcomers)	1	12.51	12.51	7.73	0.01	
log(From prior)	1	17.14	17.14	10.59	0.00	
log(Until next)	1	109.34	109.34	67.56	0.00	
log(Reports + 1)	1	14.23	14.23	8.79	0.00	
log(Left/Transferred + 1)	1	0.00	0.00	0.00	0.98	
Residuals	24004	38846.39	1.62			

Interpreting results

- Proximity to prior and subsequent organizational change
  - Proximity to a subsequent reorganization explains five times more of the variance
- Number of newcomers decreases the productivity of the existing developers
- the number of employees supervised by a developer decreases productivity
- Leaving/transferred employees have no effect: perhaps two months is not sufficient to experience resulting knowledge gaps

Digital Archeology: Sources

- **People**: Organizational Directory (LDAP) snapshots
  - Chronology: late 2001 and early 2003. Early 2004 until present: weekly extracts.
  - Attributes: personal ID, supervisor ID, department, location, phone, email
- **Mapping** VCS *login* to *LDAP id*
  - Yellow pages (NIS), weekly extracts from three clusters
    - login to LDAP attributes, name, email
  - Proprietary problem reporting system (Sablime), weekly extracts
    - login to name, email
- **Version control systems**
  - Chronology: 1990 until present
  - Attributes: login, date, file

Hypotheses

Proposition 1 *Organizational volatility* reduces quality

Proposition 2 *New experienced members would bring innovations and, therefore, find new ways to* improve quality

Proposition 3 *New inexperienced members would be more likely to* introduce defects

Proposition 4 *Outgoing members would leave gaps in the tacit knowledge, making suboptimal design and implementation decisions more likely by the remaining team. This would* increase the probability that defects will be introduced or not found *prior to release.*

Related work: replication hypotheses

Proposition 5 *Larger files will have* lower quality

Proposition 6 *Files modified by diffuse changes and files with high logical coupling will have* lower quality

Proposition 7 *Files modified by developers who have complex workflow will have* lower quality

Proposition 8 *Files modified by developers with low project experience will have* lower quality

Proposition 9 *Files modified by developers from multiple development sites will have* lower quality

Proposition 10 *Files modified by changes that are incorporated into multiple releases will have* lower quality