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**
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 than the best memory” Webb et al. [1966], Geisler [1999] propose to measure organizational change based on archival records.


- **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

5 A. Mockus Organizational Volatility and its Effects on Software Defects FSE, Santa Fe, 2010
Digital Archeology: Sources

Data Sources

IssueTracking
- MR
- Cstmr bug
- Activity
  - time
  - login
  - action
  ...

VCS
- Version
- Time
- Login
- File
- MR

NIS
- Time
- Login
- PersonNo
- email

Organization
- Time
- PersonNo
- email
- Address
- Name

6 A. Mockus  Organizational Volatility and its Effects on Software Defects  FSE, Santa Fe, 2010
Product
Development
Hierarchy
Colored by
Location
### Operationalizations of volatility

<table>
<thead>
<tr>
<th>Concept</th>
<th>Operationalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proximity in time to the organizational change</td>
<td>Time (in years) until the next and after the last change in the organization ID</td>
</tr>
<tr>
<td>Size of the reorganization</td>
<td>Number of employees leaving the organization over past two months</td>
</tr>
<tr>
<td>New recruits</td>
<td>Number of employees entering the organization over past two months</td>
</tr>
<tr>
<td>Size of the organization</td>
<td>Number of employees within the organization</td>
</tr>
<tr>
<td>Other factors</td>
<td>Product, Location, Organization ID, Developer ID</td>
</tr>
</tbody>
</table>
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 - $D_5$

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

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

Organization

Reorg. reduces quality

New, inexperienced reduce quality

Leaving lead to knowledge gaps: errors

Outcome

Defects

Size, type, ...
Defect Modeling Objective

- **Scientific**
  - Does organizational volatility affect defects after adjusting for factors known from literature?
  - **Review** related work ⇒ **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

Data Sources

- Issue Tracking
  - MR
  - Cstmr bug Activity
    - time
    - login
    - action
  ...

- VCS
  - Version
  - Time
  - Login
  - File
    - MR

- NIS
  - Time
  - Login
  - PersonNo
  - email

- Organization
  - Time
  - PersonNo
  - email
  - Address
  - Name

Measures

- Social Workflow Experience

- Change Diffusion
  - Release
  - Logical

- Geography
  - Distributed
  - Offshore
  - Mentor

- File LOC

- Organisation
  - Proximity
  - New arrivals
  - Leaving
  - Size
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

File 1

Measurement period
Prediction period
Modification

File 2

File 3

File 4

File 5

File 6

Time

Release date closest to
the 75th percentile of
files' modification times

t_1

t_2

t_3

t_4

t_5

t_6
Reproducing earlier results

Outcome

Organization
- Until Reorg
- From Reorg
- Leaving
- New
- Size

Geography
- Mentor Offshore
- Distributed

Social
- Workflow degree
- Experience

File
- LOC

Change
- Logical Deps
- Releases
- Diffusion

Defects
- 35%, reproduced
- 34%, reproduced
- 11%, reproduced
- 192%, reproduced
- 6%, reproduced

Mentor Offshore
- 38%, reproduced
- 69%, reproduced

- 15%, opposite

Distributed
- 26%, new

- -4%, new

- -15%, new

- 18%, opposite

From Reorg
- 18%, opposite

Until Reorg
- 35%, reproduced
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


21 A. Mockus Organizational Volatility and its Effects on Software Defects FSE, Santa Fe, 2010
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.

23 A. Mockus Organizational Volatility and its Effects on Software Defects FSE, Santa Fe, 2010
## Reproducing earlier results

<table>
<thead>
<tr>
<th>Class</th>
<th>Predictor</th>
<th>Effect</th>
<th>Propstns</th>
<th>Reproduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org. chng</td>
<td>Size of org.</td>
<td>38%</td>
<td>control</td>
<td>+Nagappan et al. [2008]</td>
</tr>
<tr>
<td></td>
<td>From prior (yrs)</td>
<td>-15%</td>
<td>+1</td>
<td>new result</td>
</tr>
<tr>
<td></td>
<td>Until next (yrs)</td>
<td>-4%</td>
<td>+1</td>
<td>new result</td>
</tr>
<tr>
<td></td>
<td>Left</td>
<td>26%</td>
<td>+4</td>
<td>new result</td>
</tr>
<tr>
<td></td>
<td>Newcomers</td>
<td>N/A</td>
<td>−3,2</td>
<td>new result</td>
</tr>
<tr>
<td>File</td>
<td>LOC</td>
<td>34%</td>
<td>+5</td>
<td>+various</td>
</tr>
<tr>
<td></td>
<td>Logical Deps.</td>
<td>11%</td>
<td>+6</td>
<td>+Cataldo et al. [2009], Bird et al. [2009a]</td>
</tr>
<tr>
<td>Change</td>
<td>Release Deps.</td>
<td>192%</td>
<td>+10</td>
<td>+Herbsleb and Mockus [2003b]</td>
</tr>
<tr>
<td></td>
<td>Change Diffusion</td>
<td>6%</td>
<td>+6</td>
<td>+Mockus and Weiss [2000]</td>
</tr>
<tr>
<td>Social</td>
<td>Workflow Deps.</td>
<td>35%</td>
<td>+7</td>
<td>+Cataldo et al. [2009], Bird et al. [2009a], Herbsleb and Mockus [2003b]</td>
</tr>
<tr>
<td></td>
<td>Experience (yrs)</td>
<td>18%</td>
<td>+8</td>
<td>−Mockus and Weiss [2000]</td>
</tr>
<tr>
<td>Geo.</td>
<td>Distributed</td>
<td>15%</td>
<td>+9</td>
<td>−Bird et al. [2009b], +Herbsleb and Mockus [2003a]</td>
</tr>
<tr>
<td></td>
<td>Mentor offshore</td>
<td>69%</td>
<td>+9</td>
<td>new result</td>
</tr>
</tbody>
</table>
32099 files, 7% with customer defects, 41% of deviance explained

<table>
<thead>
<tr>
<th>Class</th>
<th>Predictor</th>
<th>Est.</th>
<th>StdErr</th>
<th>p-val</th>
<th>Devnc</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>log(LOC)</td>
<td>0.43</td>
<td>0.03</td>
<td>0.00</td>
<td>2450</td>
</tr>
<tr>
<td></td>
<td>log(Logical)</td>
<td>0.25</td>
<td>0.02</td>
<td>0.00</td>
<td>978</td>
</tr>
<tr>
<td></td>
<td>log(Releases)</td>
<td>2.67</td>
<td>0.07</td>
<td>0.00</td>
<td>2331</td>
</tr>
<tr>
<td></td>
<td>log(Diffusion)</td>
<td>0.08</td>
<td>0.03</td>
<td>0.00</td>
<td>321</td>
</tr>
<tr>
<td>Chng</td>
<td>log(Workflow)</td>
<td>0.43</td>
<td>0.05</td>
<td>0.00</td>
<td>255</td>
</tr>
<tr>
<td></td>
<td>log(Experience)</td>
<td>0.28</td>
<td>0.04</td>
<td>0.00</td>
<td>13</td>
</tr>
<tr>
<td>Socl</td>
<td>log(Distributed)</td>
<td>0.14</td>
<td>0.07</td>
<td>0.04</td>
<td>41.94</td>
</tr>
<tr>
<td></td>
<td>Mentor</td>
<td>0.53</td>
<td>0.12</td>
<td>0.00</td>
<td>27.97</td>
</tr>
<tr>
<td>Geo</td>
<td>log(OrgSize)</td>
<td>0.48</td>
<td>0.06</td>
<td>0.00</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>log(From)</td>
<td>−0.40</td>
<td>0.07</td>
<td>0.00</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>log(Until)</td>
<td>−0.06</td>
<td>0.03</td>
<td>0.09</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>log(Left + 1)</td>
<td>0.33</td>
<td>0.04</td>
<td>0.00</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>log(New + 1)</td>
<td>−0.01</td>
<td>0.04</td>
<td>0.70</td>
<td>0</td>
</tr>
<tr>
<td>Class</td>
<td>Predictor</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>Size of organization</td>
<td>( \max_{l}(l, f, t-1 \leq t_0 \leq t) S(l, t_0) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time from prior change</td>
<td>( \min_{l}(l, f, t-1 \leq t_0 \leq t) P_{\text{prior}}(l, t_0) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time until next change</td>
<td>( \min_{l}(l, f, t-1 \leq t_0 \leq t) P_{\text{next}}(l, t_0) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number leaving org.</td>
<td>( \max_{l}(l, f, t-1 \leq t_0 \leq t) L(l, t) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of newcomers</td>
<td>( \max_{l}(l, f, t-1 \leq t_0 \leq t) N(l, t) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>LOC</td>
<td>Lines of non-commentary source code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Logical Deps.</td>
<td>The number of other files changed by the past MRs modifying the file: ( LD(f, t) = \mathbb{R}{f_0 : \exists mr, \exists t_1, t_2 \leq t, (f_0, mr, t_1) \land (f, mr, t_2)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Release Deps.</td>
<td>The maximum number of releases an MR is submitted to over MRs modifying the file during the \textit{measurement period}: ( R(f, t) = \max_{mr} \mathbb{R}{r : \exists t_0 \leq t, (r, mr, t_0)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Change Diffusion</td>
<td>The maximum number of files changed by an MR modifying the file during the \textit{measurement period}: ( D(f, t) = \max_{mr} \mathbb{R}{f_0 : \exists t_0 \leq t, (f_0, mr, t_0)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social</td>
<td>Workflow</td>
<td>The maximum degree of the workflow network over developers modifying the file during the \textit{measurement period}: ( W(f, t) = \max_{l} \mathbb{R}{l_0 : \exists t_1 \leq t, \exists t_2 \in [t-1, t], (l, f, t_2) \land (l_0, l, t_1)} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Years of prj. experience</td>
<td>The minimum of the years of experience over all developers modifying the file during the \textit{measurement period}.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geography</td>
<td>Distributed development</td>
<td>The number of sites that modified the file during the \textit{measurement period}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mentor Offshore</td>
<td>The maximum of the indicator that a mentor is in another site over developers modifying the file during the \textit{measurement period}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### ANOVA and regression of developer productivity ($R^2 = 0.46$)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
<th>Coef</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRID</td>
<td>1226</td>
<td>33918.00</td>
<td>27.67</td>
<td>17.10</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>SID</td>
<td>267</td>
<td>1752.13</td>
<td>6.56</td>
<td>4.05</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>log(Newcomers)</td>
<td>1</td>
<td>12.51</td>
<td>12.51</td>
<td>7.73</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>log(From prior)</td>
<td>1</td>
<td>17.14</td>
<td>17.14</td>
<td>10.59</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>log(Until next)</td>
<td>1</td>
<td>109.34</td>
<td>109.34</td>
<td>67.56</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>log(Reports + 1)</td>
<td>1</td>
<td>14.23</td>
<td>14.23</td>
<td>8.79</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>log(Left/Transferred + 1)</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>Residuals</td>
<td>24004</td>
<td>38846.39</td>
<td>1.62</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
  ✦ 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*