

# A Method to Identify and Correct Problematic Software Activity Data: Exploiting Capacity Constraints and Data Redundancies

Qimu Zheng  
Peking University

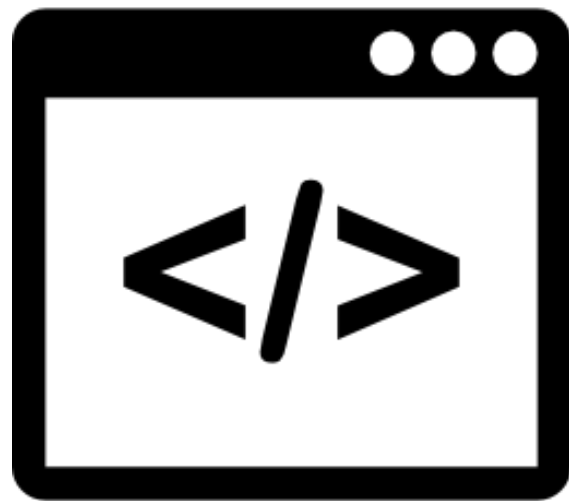
Audris Mockus  
University of Tennessee

Minghui Zhou  
Peking University





Software activity data  
are important.

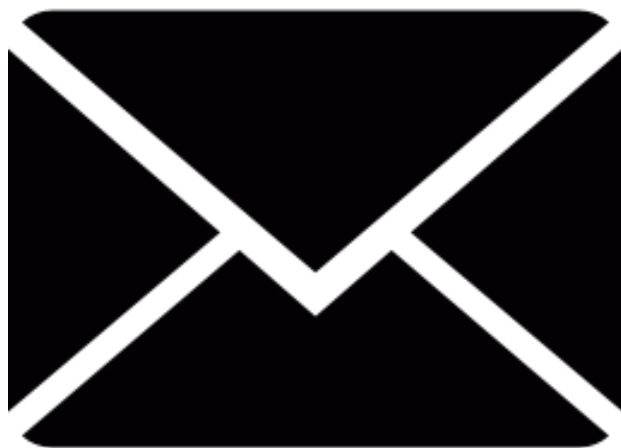


Code



Bug report

## More Available Data



Mailing list



Social media



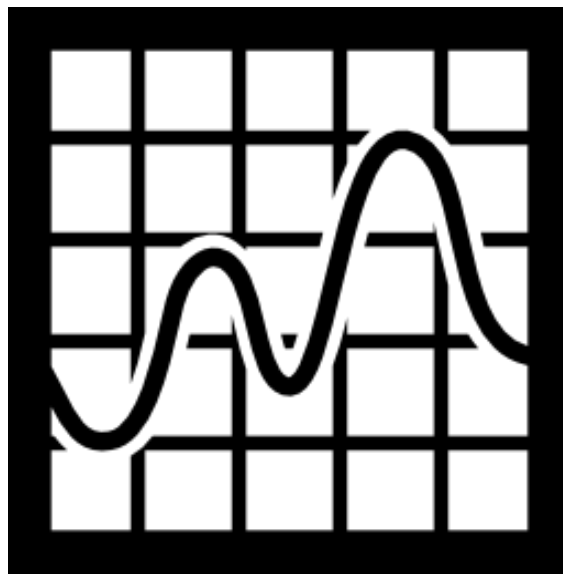
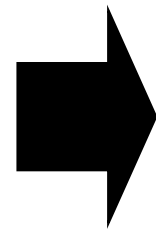
**GitHub**  
Commits



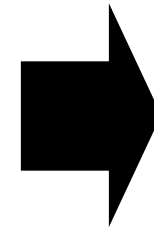
# Empirical SE Research



Software Activity Data



Statistical Model



Research Result



# Various Topics

- Measure Productivity[1]
- Duplicate Bug Report Prediction[2]
- Bug-fix Time Prediction[3]
- ...

[1] W. F. Boh, S. A. Slaughter, and J. A. Espinosa. Learning from experience in software development: A multilevel analysis.

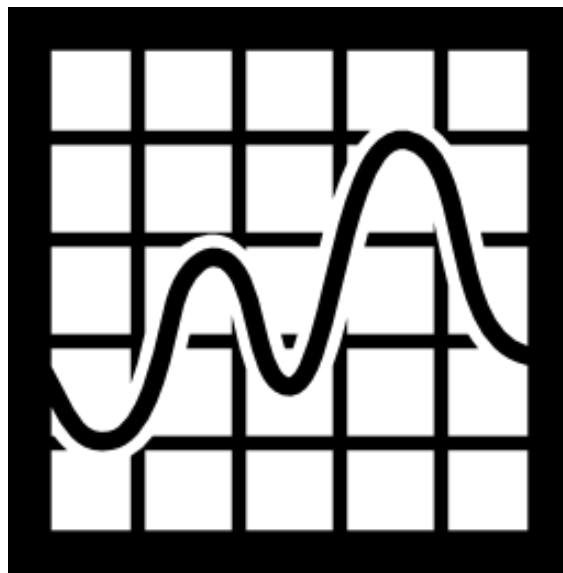
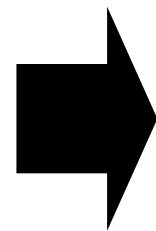
[2] Chengnian Sun; Lo, D.; Siau-Cheng Khoo; Jing Jiang, Towards more accurate retrieval of duplicate bug reports

[3] P. Bhattacharya and I. Neamtiu. Bug-fix time prediction models: Can we do better?

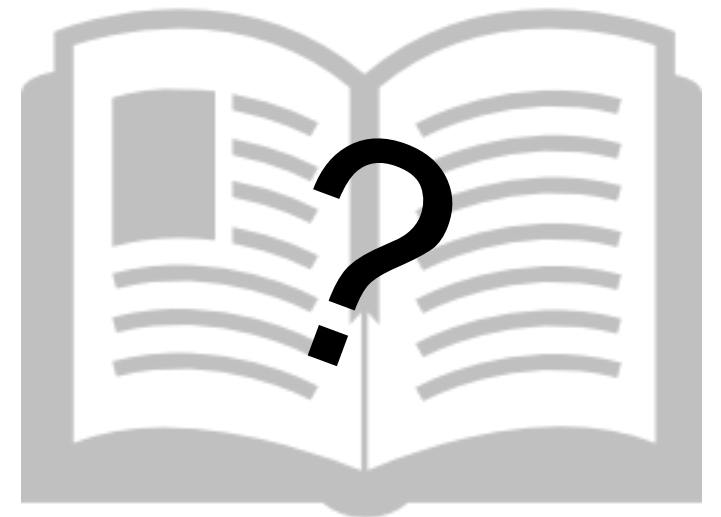
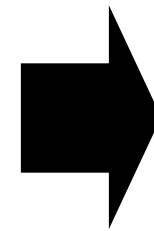
# However ...



Software Activity



Statistical Model



Research Result

1. Accomplishments  
yesterday?

2. Blockers?

3. Today's Focus?

To Do

In Progress

In Testing

We do find some  
real issues

WIS-5  
Office  
Licenses

WOP-48  
WOP-47  
1.4 System  
Requirements  
Summary

WOP-55  
WOP-44  
1.1 Product  
Vision/Scope  
Summary

WOP-57  
WOP-56  
1.2 System  
Requirements  
Summary

WOP-58  
WOP-59  
1.3 System  
Requirements  
Summary

WOP-70  
WOP-69  
1.4 System  
Requirements  
Summary

WOP-41  
VALIDATE  
FUNCTIONALITY  
OF EXISTING  
PRODUCTS

WIS-7  
Integration  
of Network  
Services

WIS-2  
WALSH  
PRINTER  
CONFIGURATION

WIS-1  
Cisco Router  
Configuration

WOP-63  
3.1 Logical  
Design  
Summary

WOP-5P  
2.3  
Architecture  
Overview

MSS-49  
Sync Test  
Consistently fails  
- Sync Issue  
018

WOP-80  
Evaluate  
DB Schema  
Tools

WOP-79  
3.2 Physical  
Diagram

WOP-36  
Infrastructure  
Page for  
Health Status  
Platform

WOP-52  
1.9 Usage  
Cases  
Summary -  
Auth. Module

WOP-35  
Validate  
Functionality  
of Existing  
Products

WOP-84  
CONFIS.  
New  
Repository

WOP-78  
CREATE  
PROTOTYPE  
AUTH.  
MODULE  
WOP-83  
1.4 - 1.5  
PROTOTYPE  
BREAKDOWN

WOP-77  
Compare  
Prototype  
Tools

WOP-71  
READ  
MFF  
DOCUMENTATION

WOP-74  
TEST  
EXISTING  
PRODUCTS

WOP-73  
READ  
AUTHENTICATION  
MODULE

RTWEB-256  
Inspection  
Data Mining

WOP-72  
SET OF  
Procedures  
To be  
Used

WOP-4  
Validate the  
Functionality  
of Existing  
Products

WIS-  
Migrate  
to Op  
36

QAWEB-  
359  
Inspection Type  
Null -  
CA 10.6

WOP-75  
Define  
Zephyr

WOP-72  
INTRO TO  
DEV.  
DEPT.



# Some real issues

- **Task completion time** is important
  - For both research and practical development
- Bug fixing time recorded in ITS is often used
- Count #bugs fixed by each dev on each day
- Experiment on official data from Mozilla

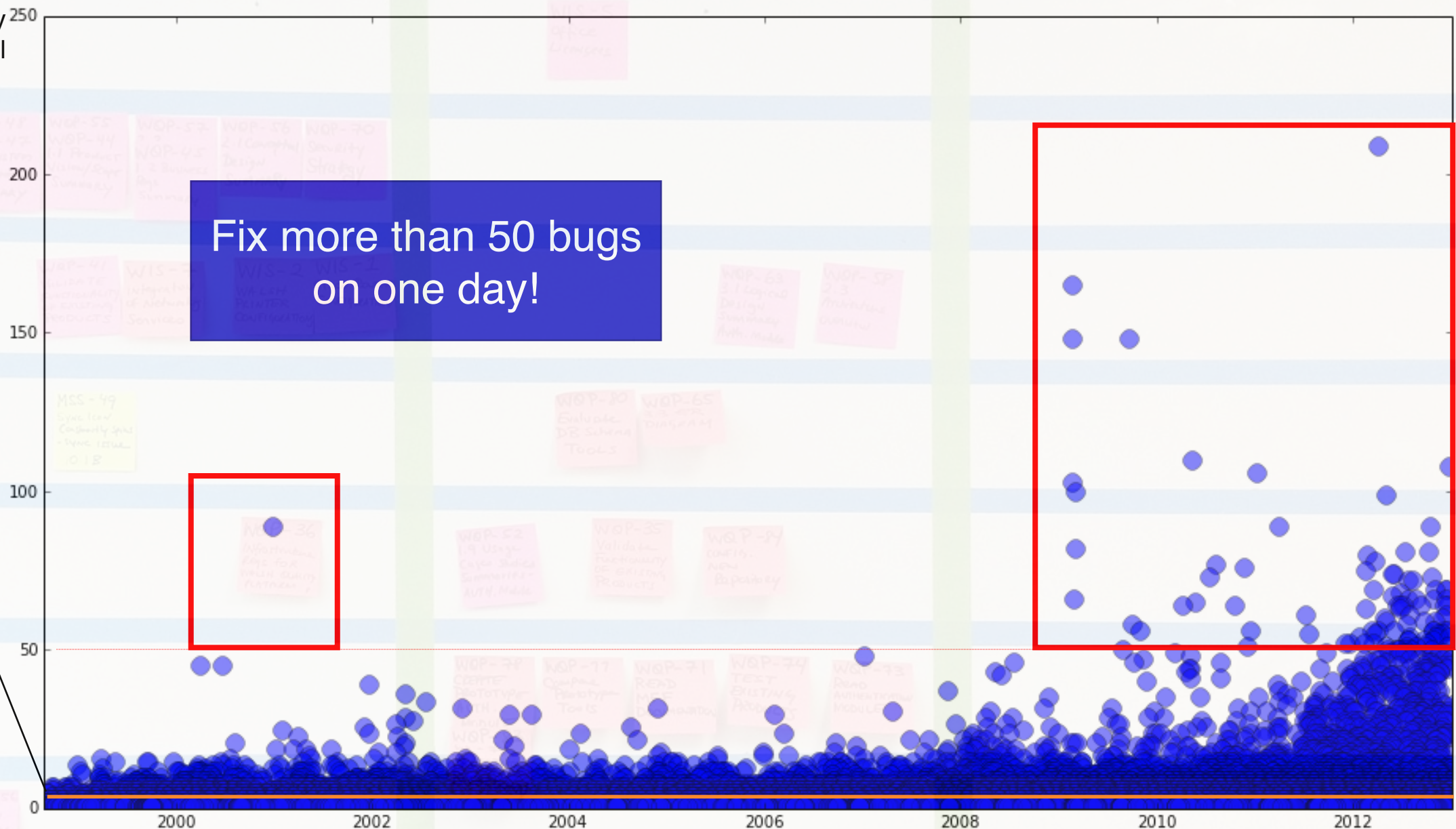


# Some real issues

#bugs fixed by  
each individual

Fix more than 50 bugs  
on one day!

Mean: 1.92



# Research on data quality?

Limited amount of work can be found.

## Research mentioning data quality?

Data quality consideration is a minority practice [1][2].

This laundry-list of potentially serious data quality issues is often either ignored or treated on a case-by-case basis in much of the extant work.[3]

[1] Michael Franklin Bosu and Stephen G. MacDonell. 2013. Data quality in empirical software engineering: a targeted review.

[2] Gernot A. Liebchen and Martin Shepperd. 2008. Data sets and data quality in software engineering.

[3] Audris Mockus. 2014. Engineering big data solutions.

# Research on Data Quality?

Limited amount of work can be found.

**Researchers love data.**

**Yet few care about  
their quality.**

Data quality consideration is a minority practice [1][2].

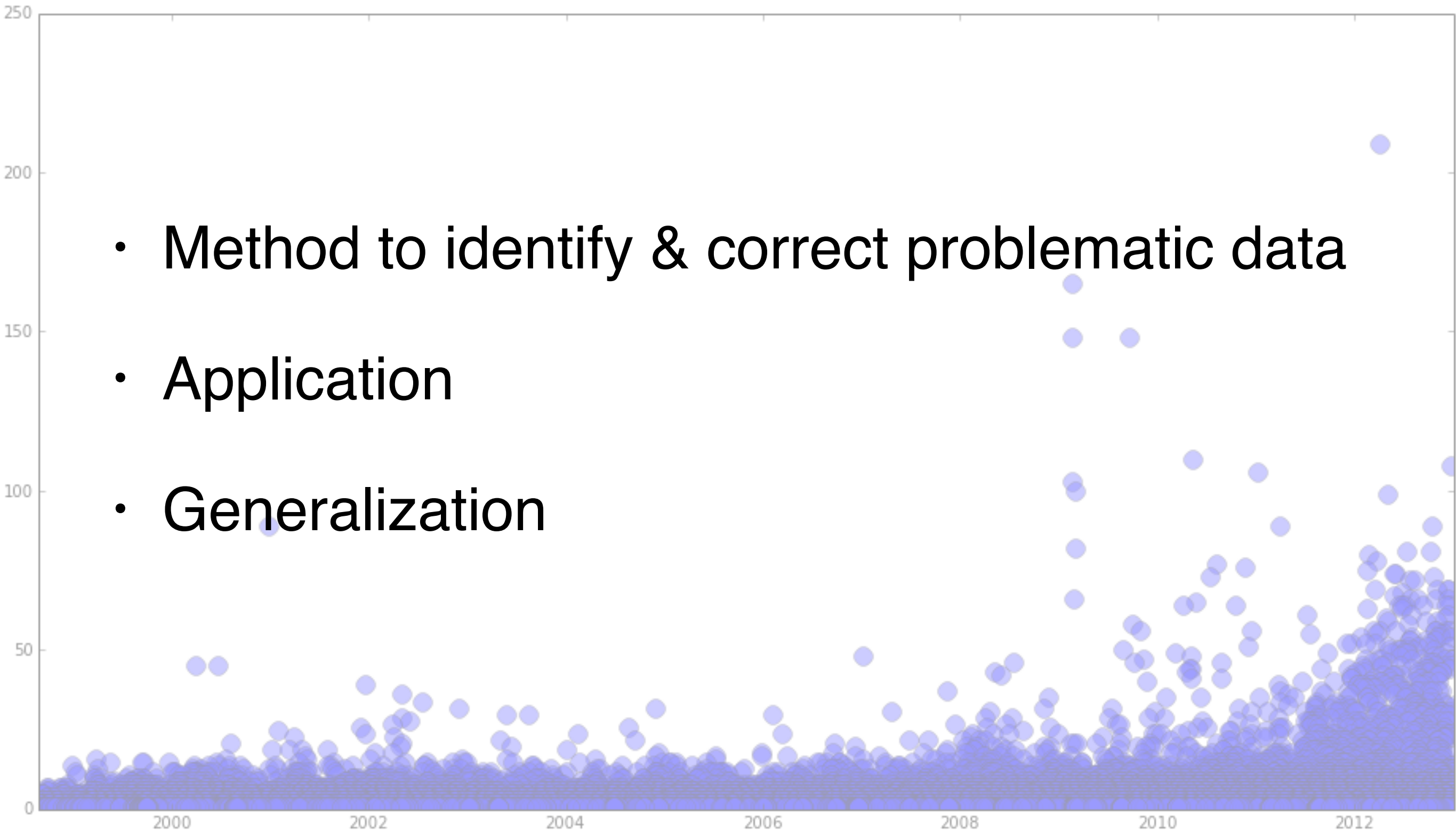
[1] Michael Franklin Bosu and Stephen G. MacDonell. 2013. Data quality in empirical software engineering: a targeted review.

[2] Gernot A. Liebchen and Martin Shepperd. 2008. Data sets and data quality in software engineering.



# So we try to fix the problem

- Method to identify & correct problematic data
- Application
- Generalization



The background features a light gray, hand-drawn style illustration. It includes a large sun in the center, two mountain ranges, and several birds in flight. The entire scene is framed by two concentric, hand-drawn circular lines.

# Before that...

## Two observations about software activity data



# • Capacity constraints



# • Data redundancies

Communication on mailing list

Code commit in VCS

Comment on issue in ITS

Issue closed in ITS

Announcement on Twitter

Time↓



# Method

for identifying and correcting  
problematic data

- **Gather data**

- Choose **primary event type** (default choice)

Data  
Redundancies

- Choose a set of **redundant event types** (some approximation)

- Obtain event times  $t_{ik}$  for task  $i$  and event  $k$ .

Capacity  
Constraints

- Use the distribution of  $t_{ik}$  to **identify problematic data**.

$isProblematic(t_{ik})$  = the likelihood that  $t_{ik}$  being incorrect.

- Obtain  $isProblematic(t_{ik})$  for each redundant observation type  $k$ .

- **Correct problematic data**. Choose observations via:

$$correct(t_i) = \begin{cases} \arg \min_{k > 1} (isProblematic(t_{ik})) & \text{if } isProblematic(t_{i1}) \\ t_{i1} & \text{if } !isProblematic(t_{i1}) \end{cases}$$

# Shorter Version

- **Gather data**
- **Choose primary event type** (default choice)
- **Choose redundant event types** (some approximation)
- **Identify problematic data**

$isProblematic(t_{ik})$  = the likelihood that  $t_{ik}$  being incorrect.

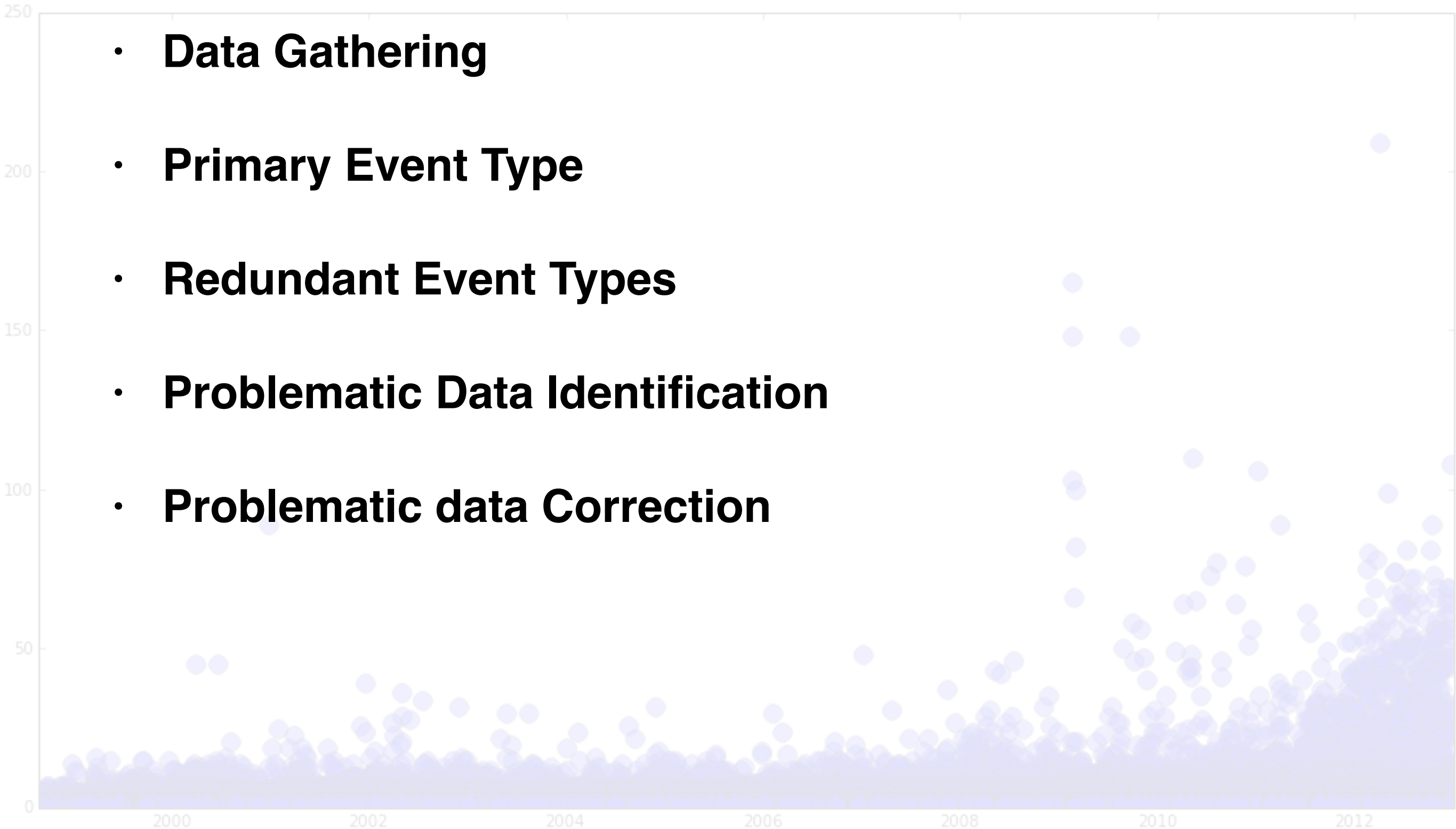
- **Correct problematic data.**

$$correct(t_i) = \begin{cases} \arg \min_{k>1} (isProblematic(t_{ik})) & \text{if } isProblematic(t_{i1}) \\ t_{i1} & \text{if } !isProblematic(t_{i1}) \end{cases}$$



# Even Shorter

- **Data Gathering**
- **Primary Event Type**
- **Redundant Event Types**
- **Problematic Data Identification**
- **Problematic data Correction**



- Data Gathering
- Primary Event Type
- Redundant Event Types
- Problematic Data Identification
- Problematic Data Correction

# Application

of the proposed method

- Data Gathering
- Primary Event Type
- Redundant Event Types
- Problematic Data Identification
- Problematic Data Correction

# Data Gathering

- Official Bugzilla dump from Mozilla (January 2013)
- All code commits data from Mozilla (February 2014)



- Data Gathering
- **Primary Event Type**
- Redundant Event Types
- Problematic Data Identification
- Problematic Data Correction

# Primary Event Type

Bug-fix time recorded in issue tracking system.

cdawson	2012-04-03 08:58:14 PDT	Status	NEW	RESOLVED
		Resolution	---	FIXED
		Last Resolved		2012-04-03 08:58:14

## Redundant Event Types?

Choose by understanding error mechanisms!

·	Data Gathering
·	Primary Event Type
·	Redundant Event Types
·	Problematic Data Identification
·	Problematic Data Correction

# Redundant Event Types

- Error mechanisms

- Development Process Tracked By Other System

- Dormant issues

- Closing issues with committed patches

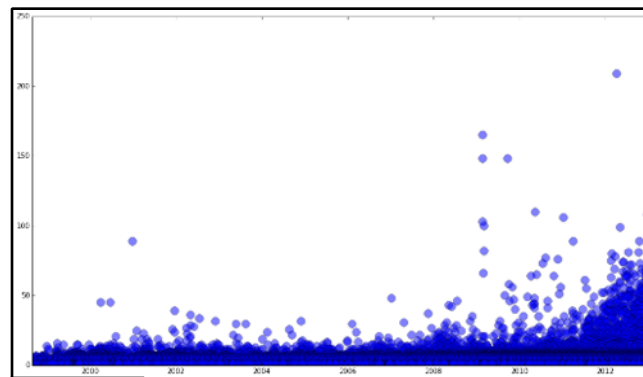
- Good substitutes:

- Last comment time

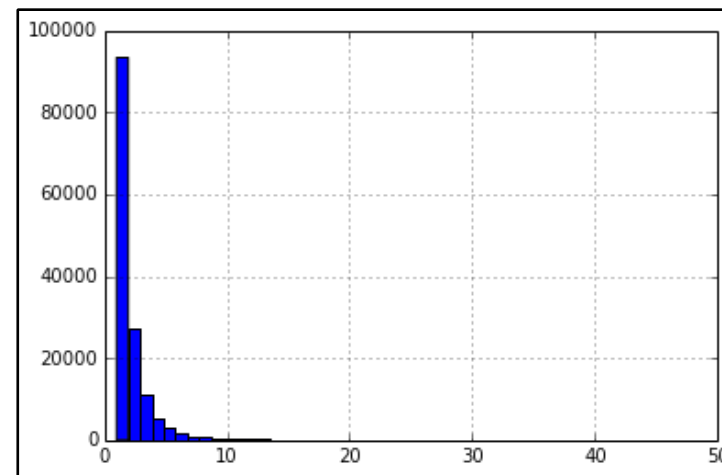
- Last code commit time

- Data Gathering
- Primary Event Type
- Redundant Event Types
- Problematic Data Identification
- Problematic Data Correction

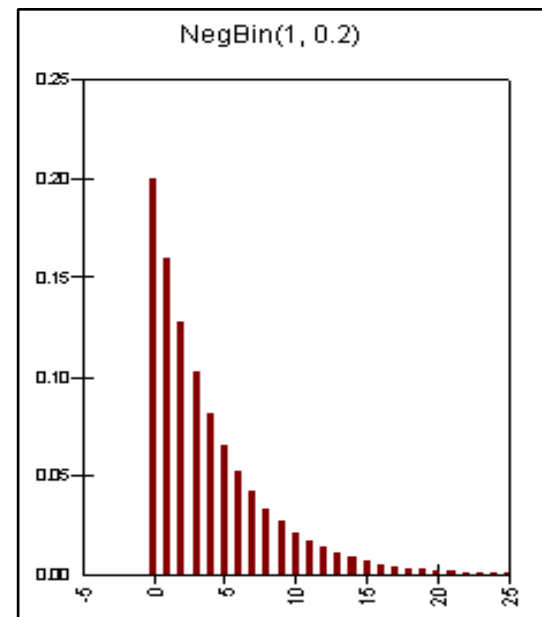
# Problematic Data Identification



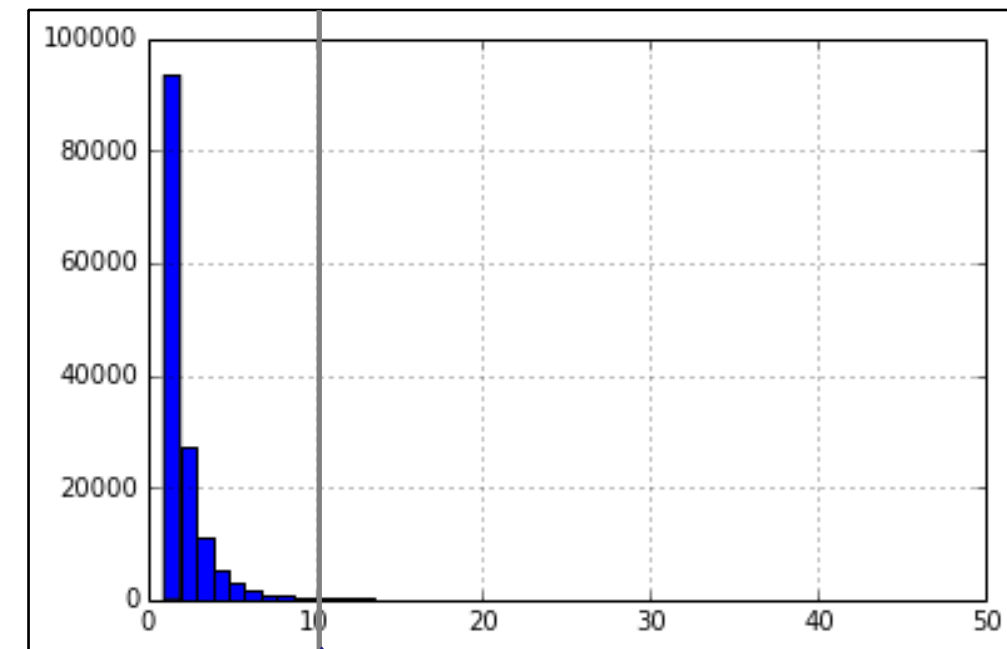
Data



Histogram



0-truncated negative binomial distribution



10 as cut-off

# Problematic Data Correction

- Available options:
  - Last comment time
  - Last commit time
- Since last commit time will be used for evaluation, we use **last comment time** for correction:

$$correct(t) = \begin{cases} \text{last comment time} & \text{if } isProblematic(\text{ITS recorded time}) \\ \text{ITS recorded time} & \text{if } !isProblematic(\text{ITS recorded time}) \end{cases}$$



Does it matter?





# Data Accuracy



# Impacts on Research

# Data Accuracy

- 16% of the issues are fixed with a link pointing to some commits in version control system (VCS)
- We take the timestamp in VCS as gold standard for evaluation

$$\text{absolute error} = |\text{timestamp} - \text{vcs timestamp}|$$

$$\text{relative error} = \frac{|\text{timestamp} - \text{vcs timestamp}|}{\text{vcs timestamp} - \text{issue creation time}}$$

**Absolute Error**

Quantile	Uncorrected	Corrected
0.50	0d 07:17:13	0d 01:08:17
0.75	1d 00:16:33	1d 11:03:00
0.80	1d 08:52:50	0d 21:21:03
0.90	5d 21:59:42	4d 12:40:42
0.99	75d 03:43:39	72d 11:18:15

**Relative Error**

Quantile	Uncorrected	Corrected
0.50	0.0205	0.0073
0.75	0.2105	0.0777
0.80	0.3700	0.1544
0.90	1.6504	0.8502
0.99	148.2818	73.3260



# Impacts on Research

Time until fix



New



Fixed

time until fixed

time

Existing research



- Summary
- Severity
- Priority
- Product
- Description



New

$\Delta t$



Fixed

Time until fixed

# Impacts on Research

$\ln(days + 1) \sim severity + \ln(attachments + 1) + reputation + \ln(assignee + 1) + \ln(depends + 1) + priority + late + \ln(comments + 1) + resolver + last_commenter$

	Estimate	p-value
(Intercept)	4.91	0.00
Critical	0.39	0.00
Major	0.64	0.00
Normal	0.80	0.00
Minor	1.02	0.00
Trivial	0.75	0.00
Enhancement	1.23	0.00
$\ln(attachments+1)$	-0.16	0.00
$\ln(depends+1)$	0.62	0.00
$\ln(assignee+1)$	0.32	0.00
Reputation	-1.04	0.00
P1	-0.22	0.00
P2	0.08	0.11
P3	0.32	0.00
P4	0.52	0.00
P5	1.33	0.00
$\ln(comments+1)$	0.54	0.00
Resolver	-0.22	0.00
Late	-0.72	0.00

	Estimate	p-value
(Intercept)	-2.23	0.02
Critical	0.28	0.01
Major	0.43	0.00
Normal	0.60	0.00
Minor	0.75	0.00
Trivial	0.75	0.00
Enhancement	1.12	0.00
$\ln(attachments+1)$	-0.12	0.00
$\ln(depends+1)$	0.41	0.00
$\ln(assignee+1)$	0.45	0.00
Reputation	-0.52	0.00
P1	-0.09	0.05
P2	0.20	0.00
P3	0.43	0.00
P4	0.49	0.00
P5	0.85	0.00
$\ln(comments+1)$	1.08	0.00
Resolver	-0.21	0.00
Late	-0.20	0.00

# Impacts on Research

$\ln(days + 1) \sim severity + \ln(attachments + 1) + reputation + \ln(assignee + 1) + \ln(depends + 1) + priority + late + \ln(comments + 1) + resolver + last_commenter$

R2: 0.381 => 0.452

Predictors: 4 significancy changes

Correction of data makes a substantial difference.

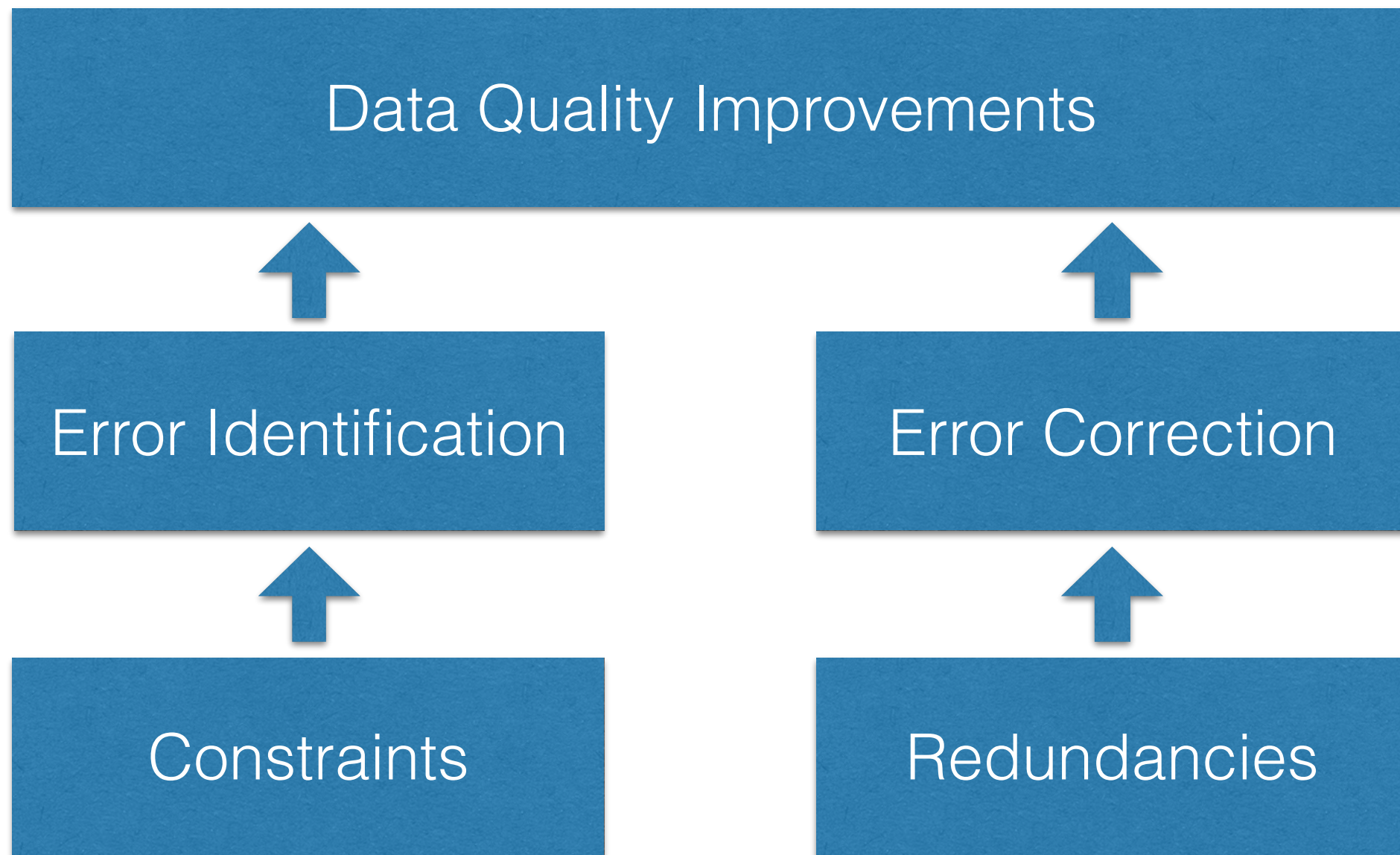
	Estimate	Std. Error
(Intercept)	4.91	0.00
Critical	0.39	0.00
Major	0.64	0.00
Normal	0.80	0.00
Minor	1.02	0.00
Trivial	0.75	0.00
Enhancement	1.23	0.00
In(depends+1)	0.62	0.00
In(assignee+1)	0.32	0.00
Reputation	-1.04	0.00
P1	-0.22	0.00
P2	0.08	0.11
P3	0.32	0.00
P4	0.52	0.00
P5	1.33	0.00
In(comments+1)	0.54	0.00
Resolver	-0.22	0.00
Late	-0.72	0.00

	Estimate	Std. Error
(Intercept)	2.29	0.02
Critical	0.28	0.01
Major	0.43	0.00
Normal	0.60	0.00
Minor	0.75	0.00
Trivial	0.75	0.00
Enhancement	0.75	0.00
In(depends+1)	0.41	0.00
In(assignee+1)	0.45	0.00
Reputation	-0.52	0.00
P1	-0.09	0.05
P2	0.20	0.00
P3	0.43	0.00
P4	0.49	0.00
P5	0.85	0.00
In(comments+1)	1.08	0.00
Resolver	-0.21	0.00
Late	-0.20	0.00



# Generalization

# Generalization



# Generalization

Exceptionally “Productive” Individuals  
(Based on Issue Report Events)

Date	User ID	Count
2012-10-01	452624	542
1999-11-22	4415	277
2011-06-24	12809	116
2009-12-16	24572	110
2012-01-27	148348	93
2012-10-12	384312	90
2011-12-14	24572	87
2010-10-13	164048	87
2012-06-01	24572	86
2000-07-08	41	86

Exceptionally “Productive” Individuals  
(Based on Code Commit Events)

Date	User ID	Count
2013-03-21	Bobby Holley	1160
2013-08-22	Ms2ger	1029
2013-02-25	Gregory Szorc	1024
2014-01-27	B2G Bumper Bot	998
2012-08-04	Ms2ger	991
2013-07-24	Ms2ger	986
2013-01-08	ffxbld	981
2011-07-21	ffxbld	964
2013-08-06	ffxbld	945
2013-02-20	ffxbld	907



Thank you!