Release Pattern Discovery via Partitioning: Methodology and Case Study

Abram Hindle, Michael W. Godfrey, Richard C. Holt

Software Architecture Group
David R. Cheriton School of Computer Science
University of Waterloo
Canada

{ahindle,migod,holt}@cs.uwaterloo.ca

Introduction

- Methodology for analyzing revisions around releases
- Discover project behaviour
- Automated Process Extraction from change histories (version control)
- Release Time is the end and start of an iteration.

Introduction

- Value of Process Discovery
  - Verify what programmers are doing
  - Extract successful processes
  - Avoid unsuccessful processes
  - Do not have to rely on witnesses to the development

Introduction

- *For each class of revision, does the frequency of those revisions increase (or decrease) preceding (or following) the time of the release?*

Terminology

- Revision
- Major and Minor Releases
- Revision Classes
  - Source,Test,Build, and Documentation Revisions
- Release Pattern

Methodology

- Extract
- Partition
- Aggregate
- Analyze
  - STBD Notation

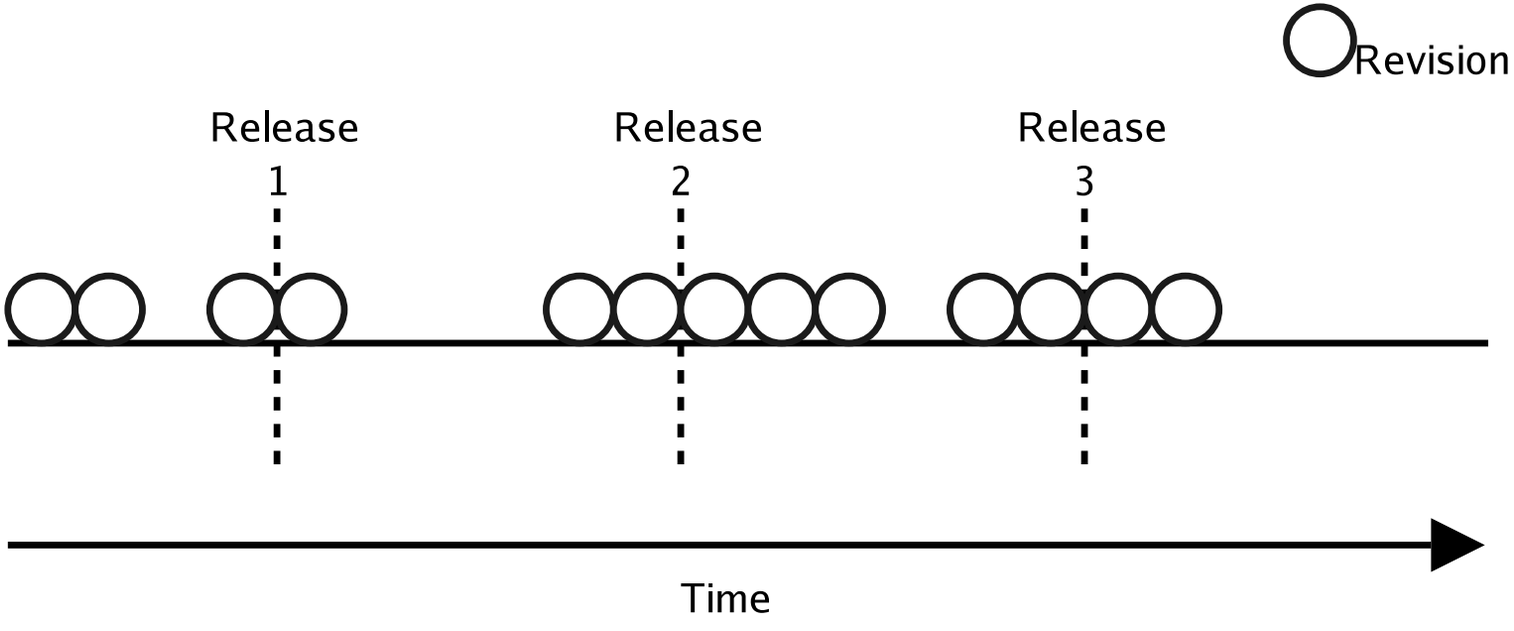

Figure 1: Revisions and releases over time. Extract the revisions

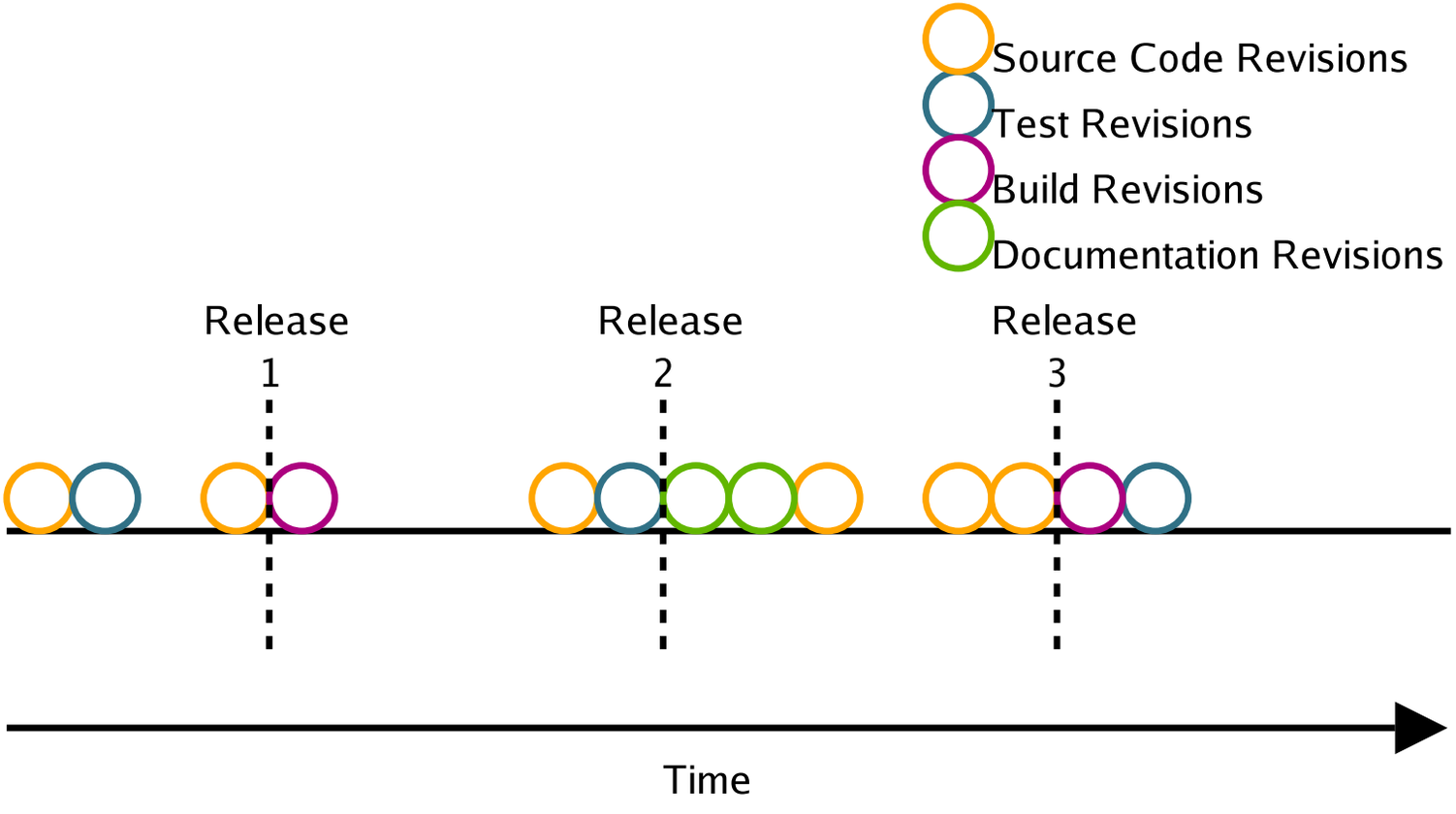

Figure 2: Partitioned revisions and releases over time

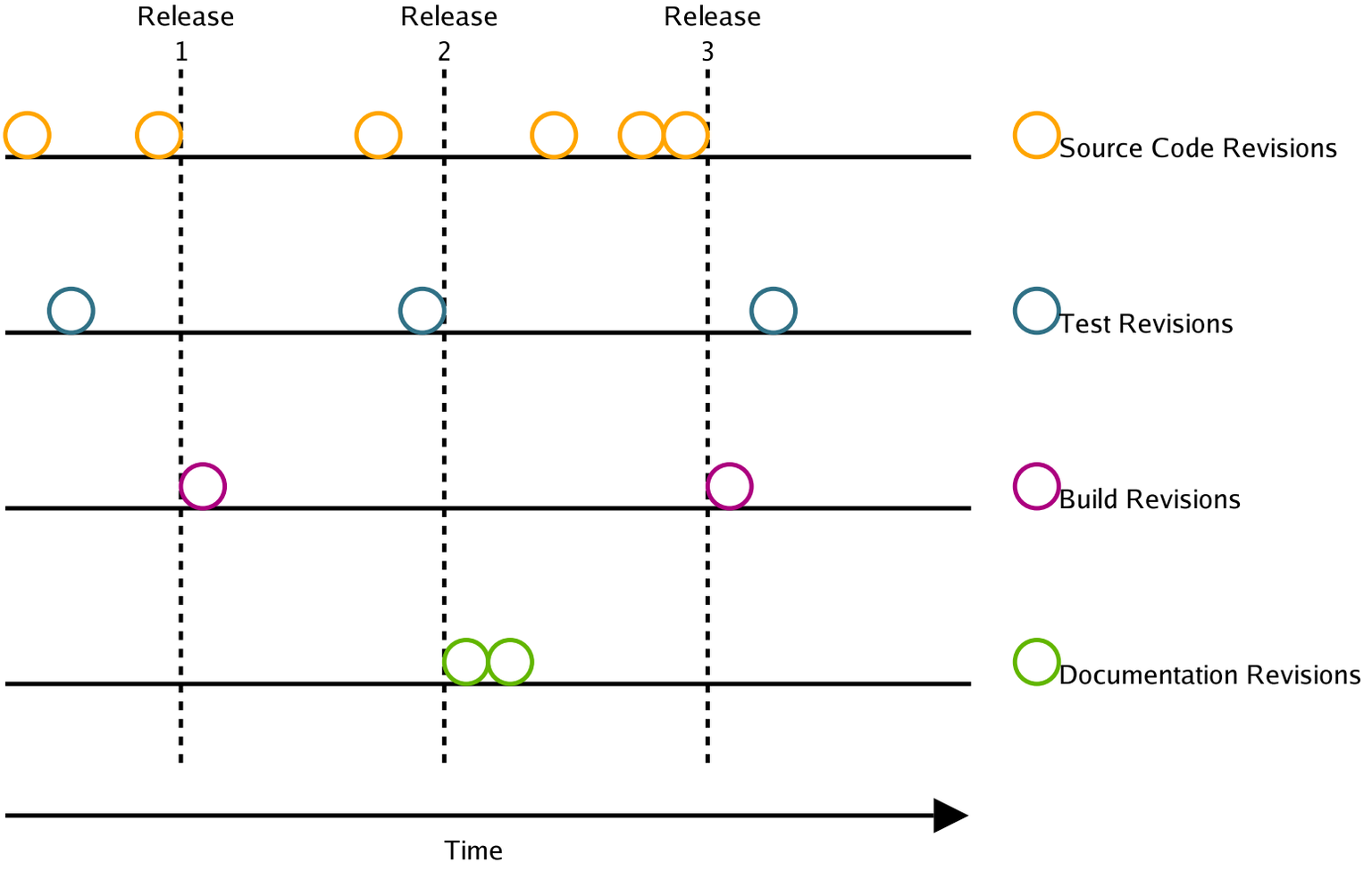

Figure 3: Partitioned revisions and releases over time, separated

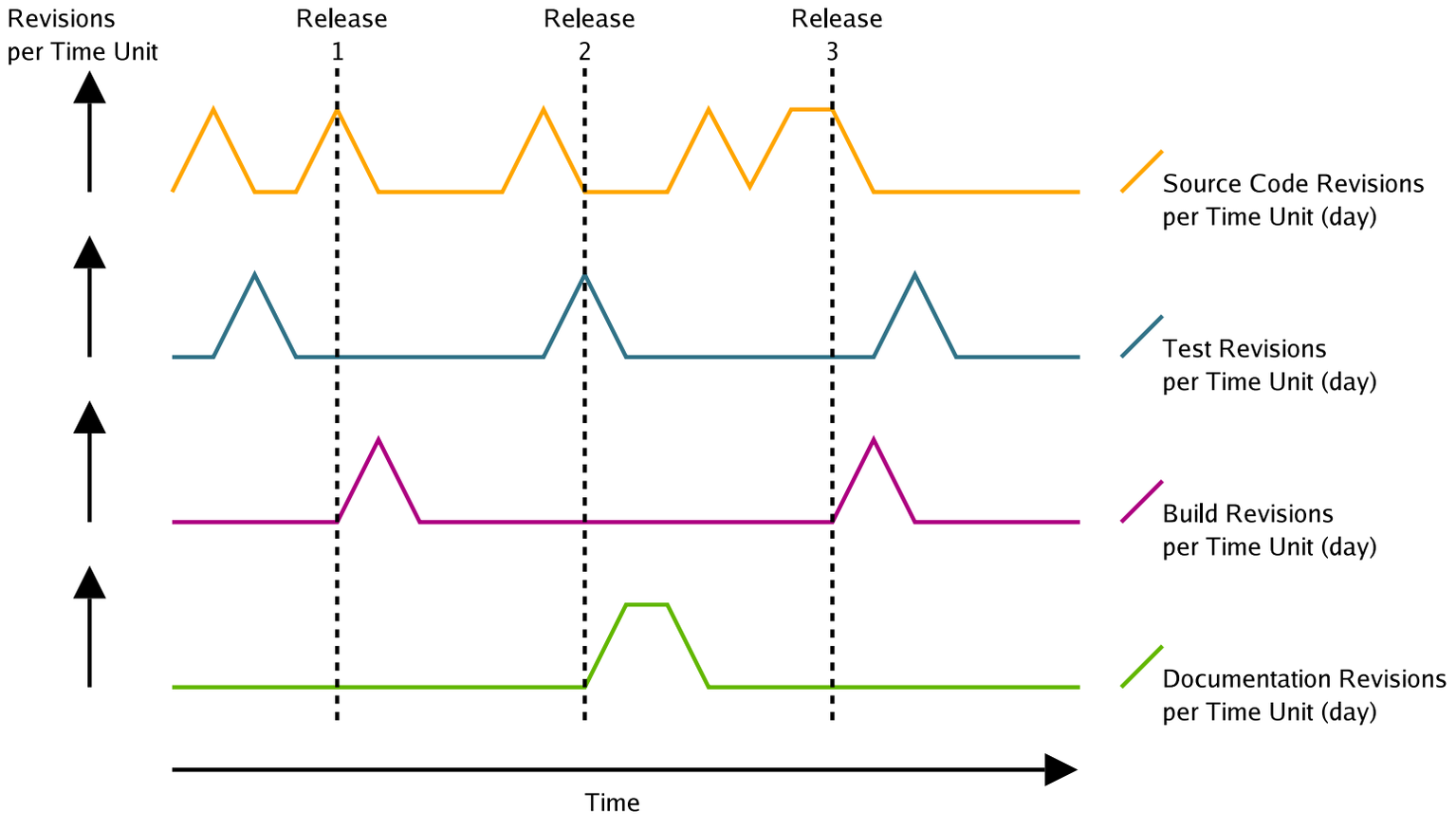

Figure 4: Partitioned revisions aggregated per day

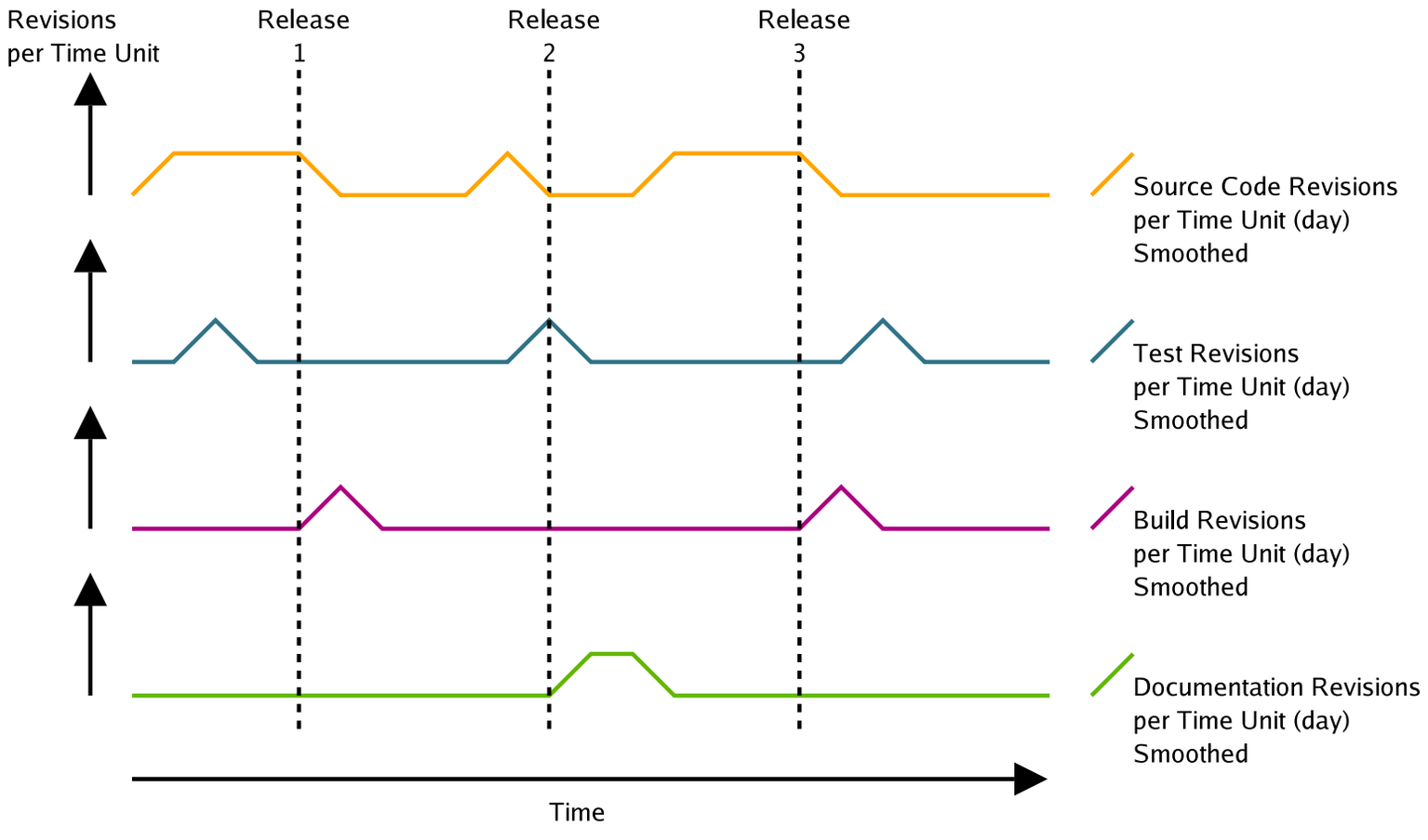

Figure 5: Partitioned revisions aggregated per day and smoothed

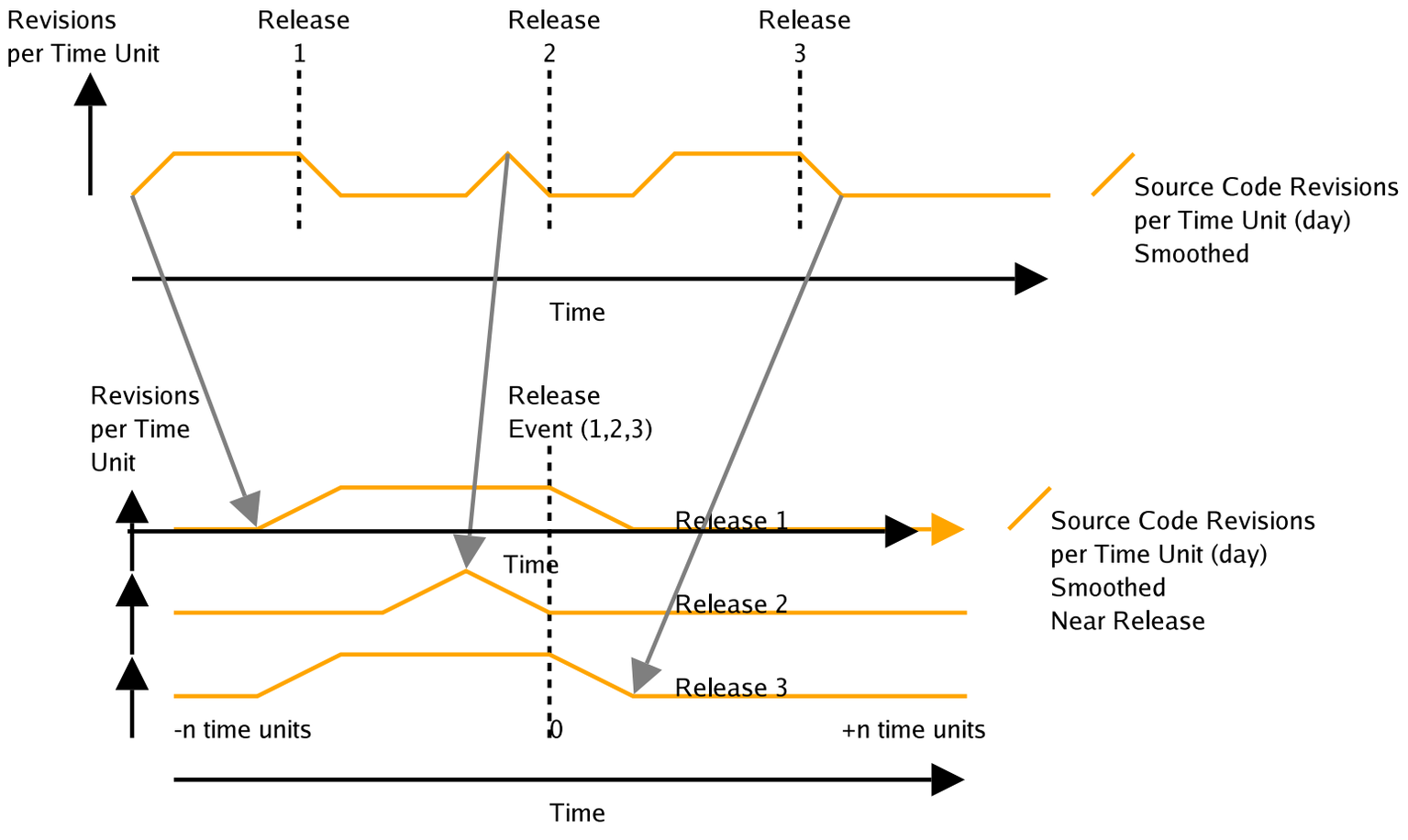

Figure 6: Select the revisions around release times

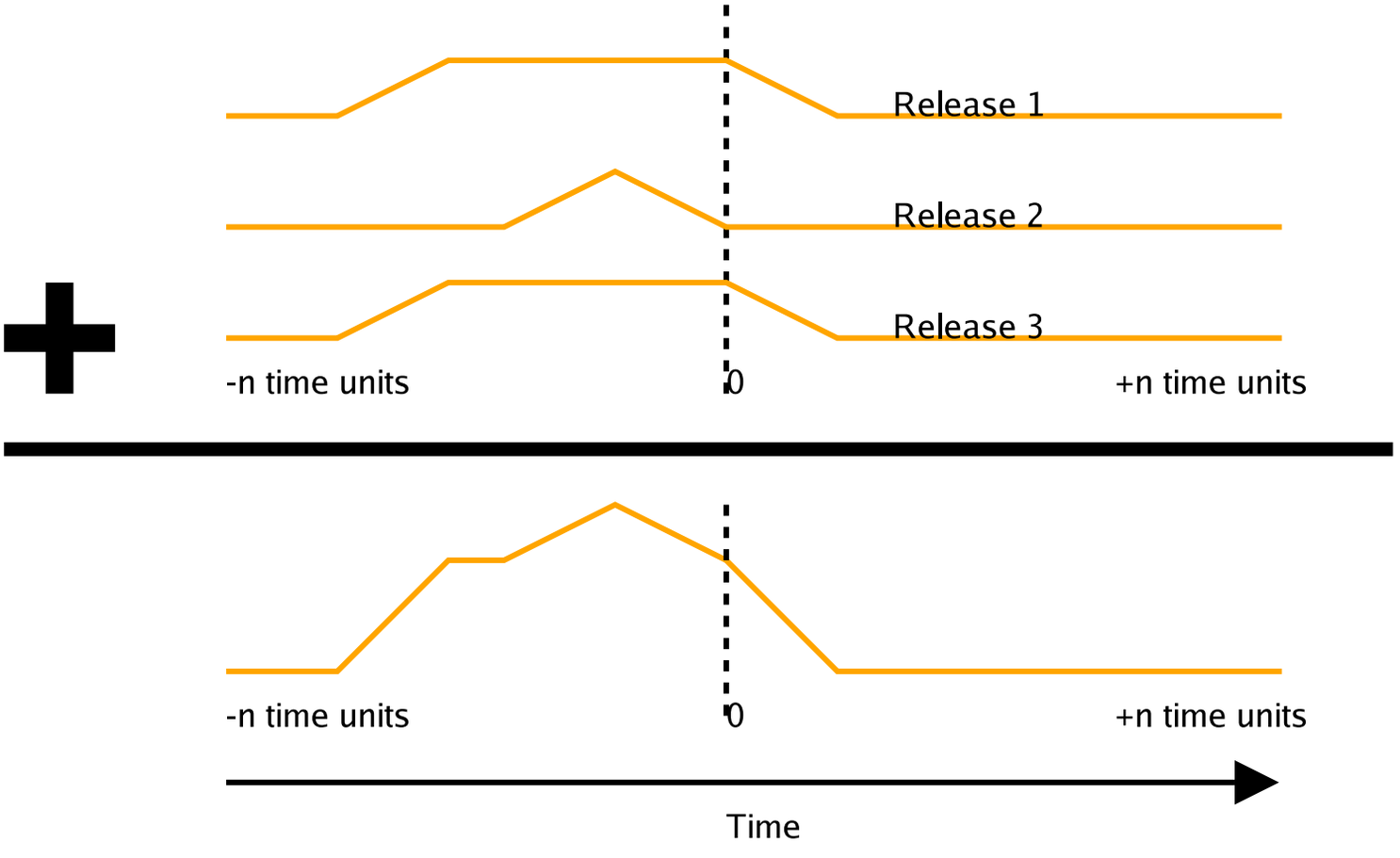

Figure 7: Aligned revisions aggregated

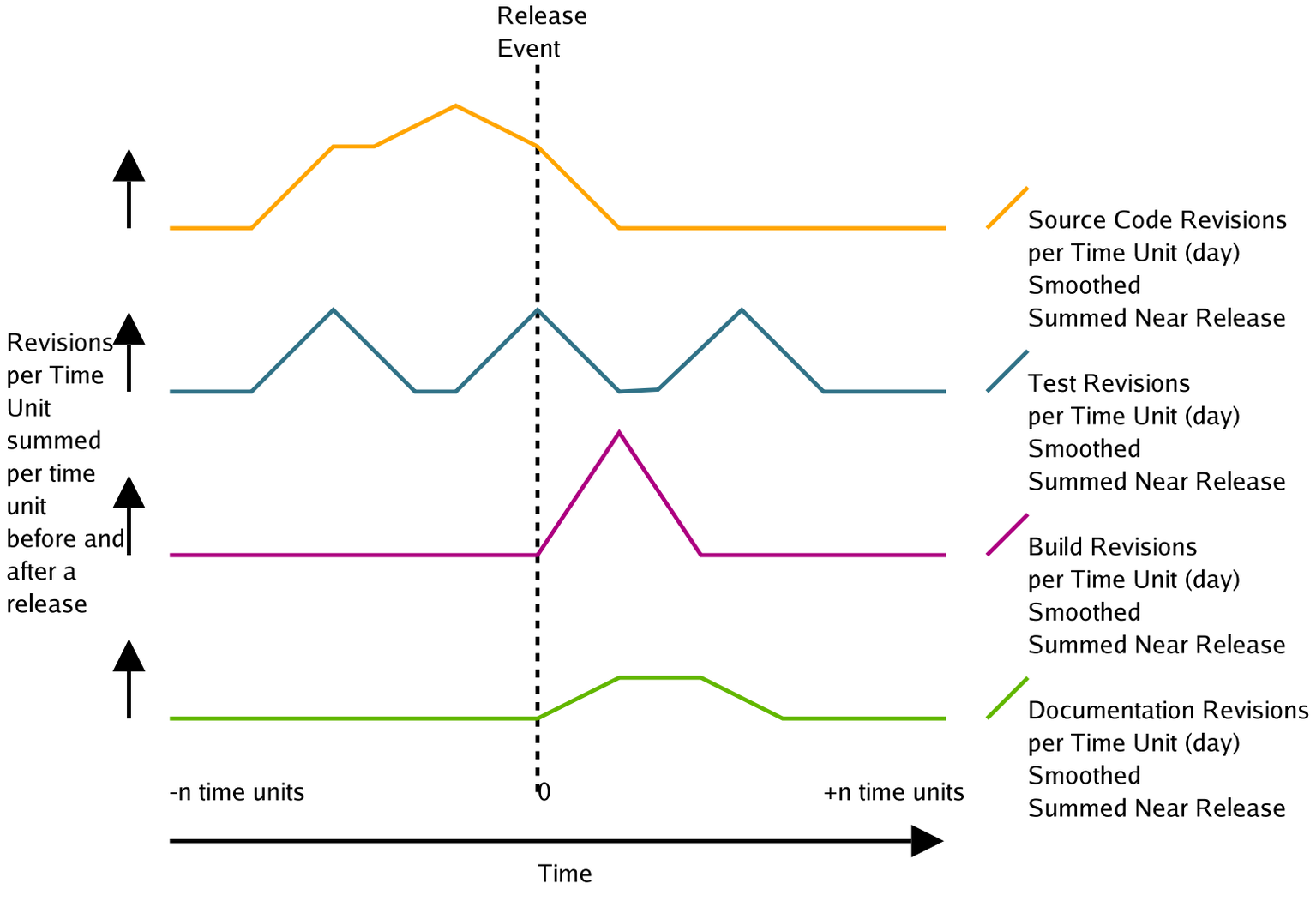

Figure 8: Align and aggregate revisions of each class

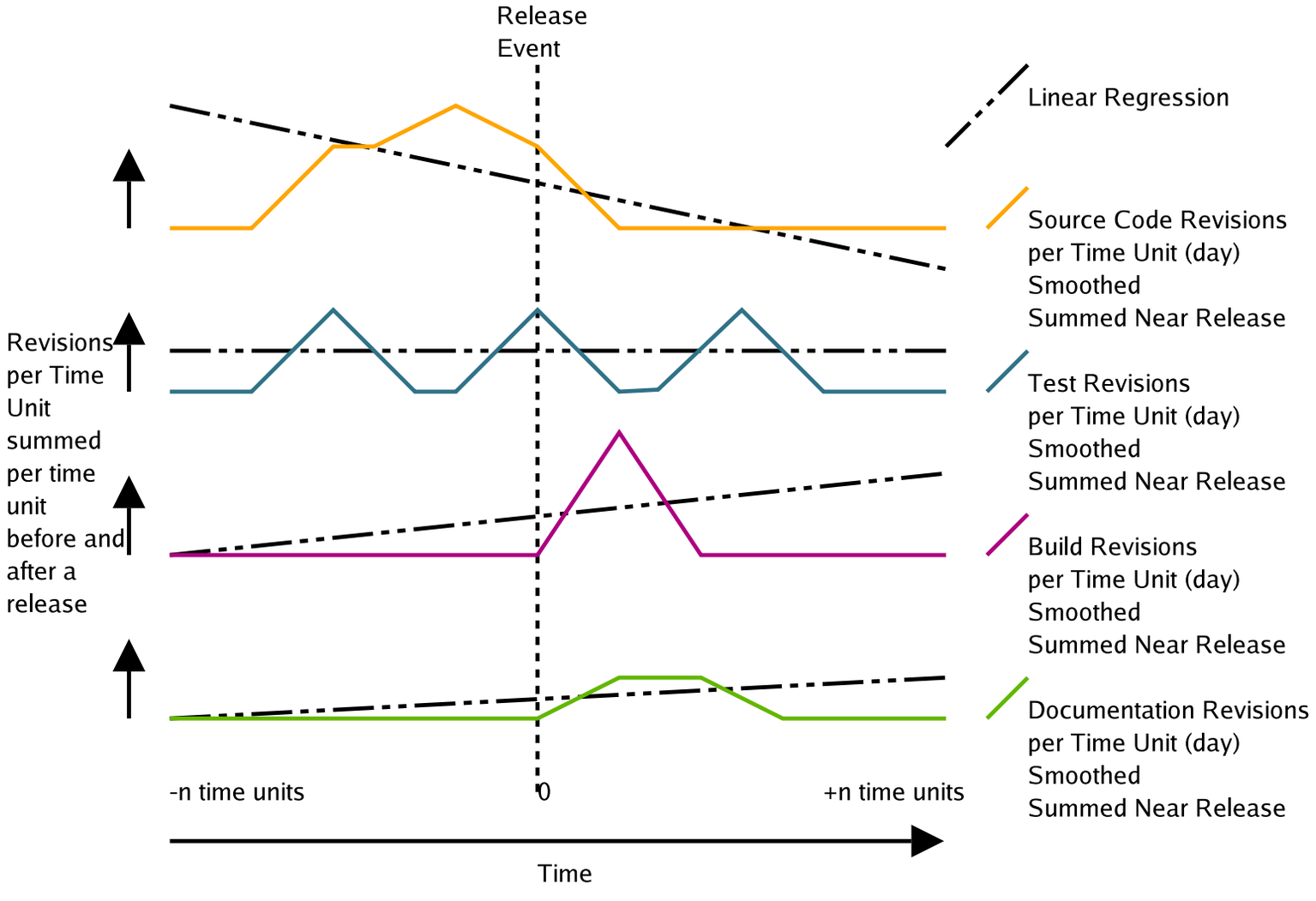

Figure 9: Analysis: averages and linear regressions

STBD Notation

- Shows *relative revision frequency* around a release
- Shows *slope of the linear regression* around a release
- Prefixes: Source S, Test T, Build B and Docs D
  - \+ more before a release or positive slope
  - \- more after a release or negative slope
  - = equal before and after a release or flat slope
  - ? undecided
- Examples: S+T+B+D+, S-T-B-D-, S+T+B-D=

Case Study of MySQL

- Popular Open Source RDBMS
- Evaluated parallel branches: 3.23, 4.0, 4.1, 5.0, 5.1
- BitKeeper repository, used bt2csv to extract change log and revision information
- Aggregated per day
- 33 Major Releases across all branches and 563 Minor releases across all branches.
- Analyzed with bt2csv, HiraldoGrok, GNUPlot, R

Case Study of MySQL

- Extraction
  - Extract both revisions and release events
  - Extraction Tools for Revisions
    - softChange - For CVS and the Schema of extracted data
    - bt2csv - Extractor BitKeeper, extracts into a softChange schema

Case Study of MySQL

- Extraction
  - Extract Releases
    - Manual
    - VCS Tags, Changelogs, Manuals, date-stamps in FTP repositories.
    - The MySQL manual contained release info

Project	Source	Test	Build	Doc
MySQL 3.23	4 220	1 410	421	21
MySQL 4.0	11 593	4 936	1 033	34
MySQL 4.1	31 451	16 430	2 990	88
MySQL 5.0	45 946	26 373	3 908	105
MySQL 5.1	52 897	31 389	4 772	122
Total	259 822	104 528	24 095	4 137

Table 1: Total Number of Revisions per class

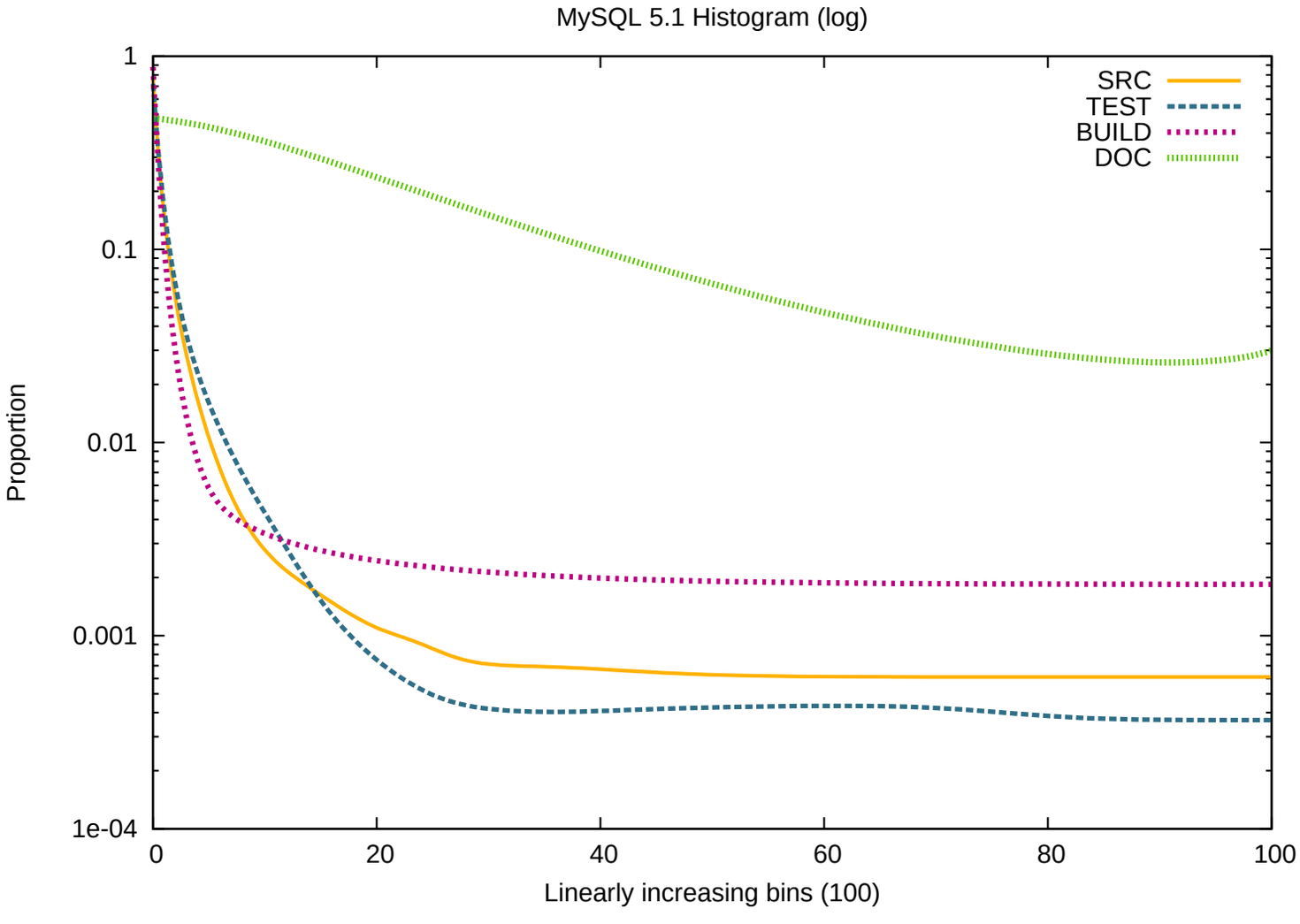

Figure 10: Distribution of revision classes for MySQL 5.1

Project	Major	Minor	All
MySQL 3.23	S-T+B-D+	S+T+B+D+	S+T+B+D+
MySQL 4.0	S+T+B-D+	S+T?B?D+	S+T?B?D+
MySQL 4.1	S+T+B-D=	S+T+B?D+	S+T+B?D+
MySQL 5.0	S+T+B-D+	S+T+B?D+	S+T+B?D+
MySQL 5.1	S+T+B-D+	S+T-B+D+	S+T-B?D+

Table 2: Summary of revision frequencies before and after release using majority voting where '?' means no majority (voting over intervals of 7, 14, 31 and 42 days)

Project	Major	Minor	All
MySQL 3.23	S-T+B-D+	S+T+B+D+	S+T+B+D+
MySQL 4.0	S+T+B-D+	S+T?B?D+	S+T?B?D+
MySQL 4.1	S+T+B-D=	S+T+B?D+	S+T+B?D+
MySQL 5.0	S+T+B-D+	S+T+B?D+	S+T+B?D+
MySQL 5.1	S+T+B-D+	S+T-B+D+	S+T-B?D+

Table 3: Summary of revision frequencies before and after release using majority voting where '?' means no majority (voting over intervals of 7, 14, 31 and 42 days)

Project	Major	Minor	All
MySQL 3.23	S-T+B-D+	S+T+B+D+	S+T+B+D+
MySQL 4.0	S+T+B-D+	S+T?B?D+	S+T?B?D+
MySQL 4.1	S+T+B-D=	S+T+B?D+	S+T+B?D+
MySQL 5.0	S+T+B-D+	S+T+B?D+	S+T+B?D+
MySQL 5.1	S+T+B-D+	S+T-B+D+	S+T-B?D+

Table 4: Summary of revision frequencies before and after release using majority voting where '?' means no majority (voting over intervals of 7, 14, 31 and 42 days)

Project	Major	Minor	All
MySQL 3.23	S-T+B-D+	S+T+B+D+	S+T+B+D+
MySQL 4.0	S+T+B-D+	S+T?B?D+	S+T?B?D+
MySQL 4.1	S+T+B-D=	S+T+B?D+	S+T+B?D+
MySQL 5.0	S+T+B-D+	S+T+B?D+	S+T+B?D+
MySQL 5.1	S+T+B-D+	S+T-B+D+	S+T-B?D+

Table 5: Summary of revision frequencies before and after release using majority voting where '?' means no majority (voting over intervals of 7, 14, 31 and 42 days)

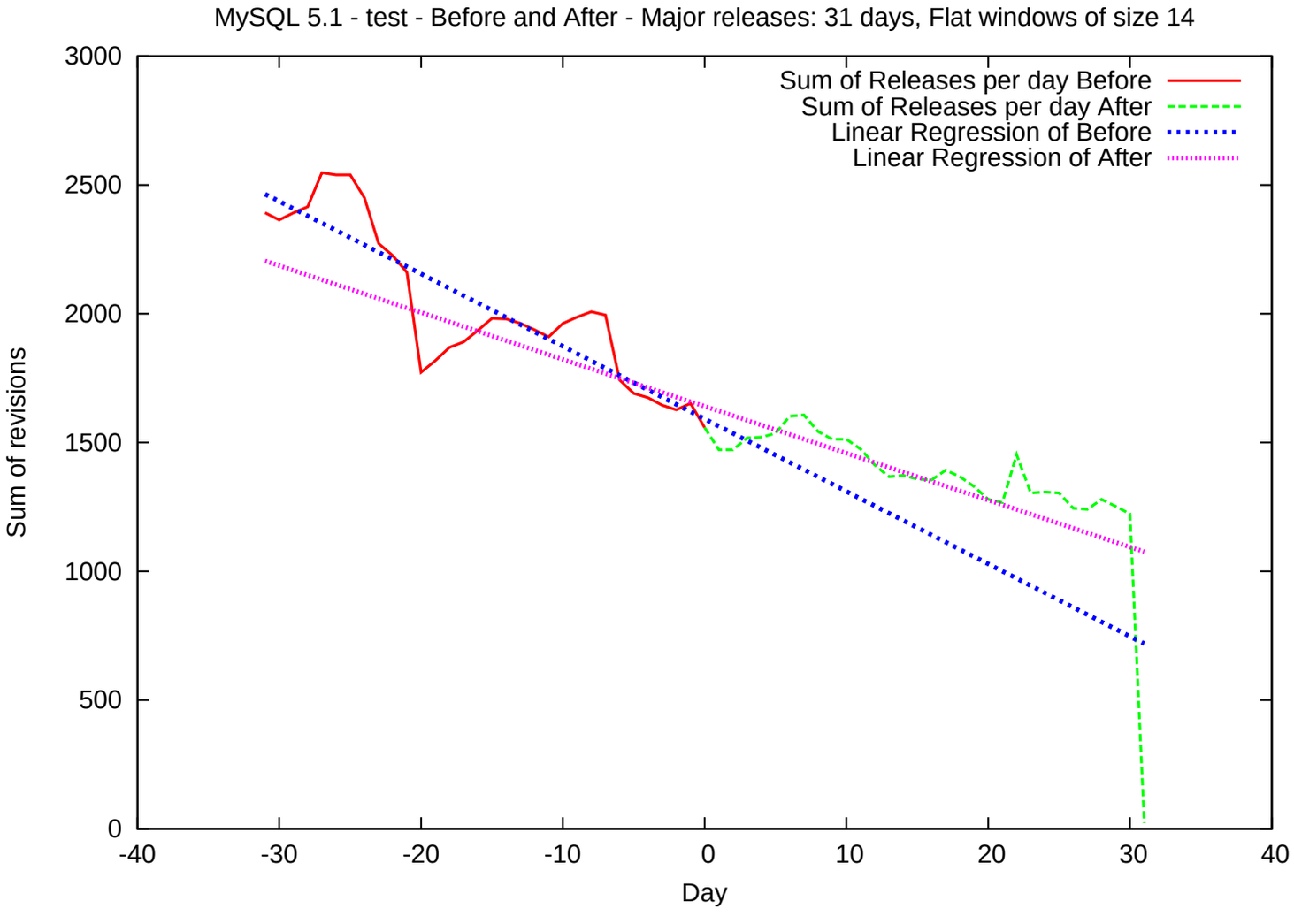

Figure 11: Windowed plot of Test revisions

Project	Before	After	Both
MySQL 3.23	S-T-B+D+	S+T-B+D=	S+T-B+D+
MySQL 4.0	S+T-B-D-	S+T-B+D=	S+T-B+D-
MySQL 4.1	S+T-B-D+	S-T-B+D+	S-T-B+D+
MySQL 5.0	S+T-B-D-	S-T-B+D-	S-T-B+D+
MySQL 5.1	S+T-B-D-	S+T-B-D+	S+T-B+D+

Table 6: Linear Regressions of daily revisions class totals:
+ indicates a positive slope, - indicates a negative slope,
= indicates a slope near 0 (Major releases, 42 day interval)

Project	Before	After	Both
MySQL 3.23	S-T-B+D+	S+T-B+D=	S+T-B+D+
MySQL 4.0	S+T-B-D-	S+T-B+D=	S+T-B+D-
MySQL 4.1	S+T-B-D+	S-T-B+D+	S-T-B+D+
MySQL 5.0	S+T-B-D-	S-T-B+D-	S-T-B+D+
MySQL 5.1	S+T-B-D-	S+T-B-D+	S+T-B+D+

Table 7: Linear Regressions of daily revisions class totals:

+ indicates a positive slope, - indicates a negative slope,

= indicates a slope near 0 (Major releases, 42 day interval)

Case Study of MySQL

- Notable behavior
  - Frequencies of S+T+D+ were common for most Major and Minor Releases
  - Frequency of B- was common for Major Releases
  - MySQL probably doesn’t follow a test-first methodology (S+T- in slope across release)
    - S+T- does not imply test first
  - Consistency and Inconsistency across branches

Future Work

- Characterize the whole process instead of the just release time
- More analysis techniques
- Analyze the difference between Major and Minor releases
- Study more projects make broader more global generalizations

Summary

- Provided a methodology to generalize about behaviour
- Characterized release time behaviour via partitioning
- Provided an initial step towards automated process extraction
- Showed that partitioning revisions allows for a more process based analysis
- Characterized release patterns of MySQL

Thank you

- Any Questions?

Project	Major	Minor	All
MySQL 3.23	2	68	70
MySQL 4.0	4	110	114
MySQL 4.1	4	110	114
MySQL 5.0	4	110	114
MySQL 5.1	4	110	114
Total	33	563	595

Table 8: Number of Major and Minor Revisions in each branch (note that MySQL 4.0 to 5.1 share the same releases)