How Code Reviews Relate to the Bugs in Software

A Hyrax Case Study

- Shivani Gadipe
- Tu Lam
- Alexander Yang Rhoads

Introduction

- We’ll be evaluating the correlation that code reviews counts, active reviewer numbers, and code depth may have with number of bugs within a release.

- The motivation for this study is that an examination could provide data to improve code review practices.

- Due to its public availability, the open-source project Hyrax will serve as our case study.

Research Question

- RQ1: Does the number of code reviews, reviewers, or review depth of a project have a correlation with the number of bugs found after release?

Methodology

- The dataset consists of the repository metadata of the open-source Hyrax repository.
  - Public history of issues, pull requests, and comments.
  - As a result, this study has access to the code reviews of each pull request, the number of reviewers, and the amount of comments within each review.
  - The study will focus exclusively on closed pull requests and issues.

- The study will track the frequency of code reviews, the number of active reviewers, the depth of reviews, and the bug occurrences across five distinct release lifecycles of Hyrax (V1->V2, V2->V3, V3->V4, V4->V5, and V5->V5.04 [latest]).
  - Tracked data is then plugged into the evaluation metrics.

- Data regarding the frequency of code reviews and the amount of reviews are directly fetched using the GitHub API.

Quantitative Parameters

- **Code Review Frequency:** Number of individual code reviews. Tracked per release of Hyrax.

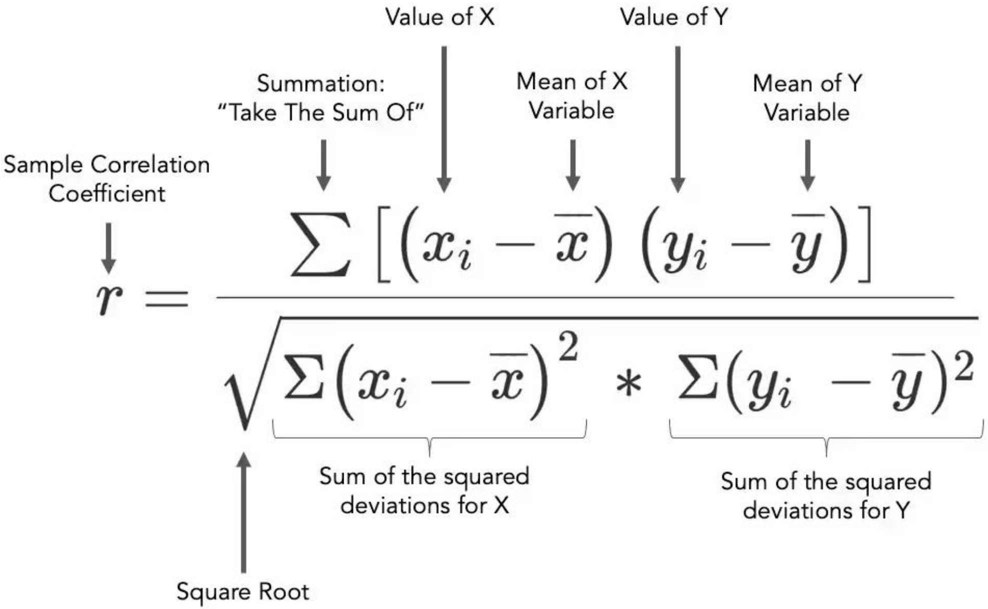
- **Reviewer Count**: Number of distinct reviewers per Hyrax release.

- **Code Review Depth:** The number of changes requested by a reviewer, divided by changes made by the PR owner. Higher depth indicates more detailed feedback.

- **Bug Occurrence:** Amount of closed bug tickets during the lifecycle of one release of Hyrax going into another.

Evaluation Metrics (Correlation Coefficient)

- Pearson Product-Moment Correlation
  - Symbolized as the r-value.
  - Meant to measure the strength of a linear relationship between two variables.
  - The r-value is between -1 and 1
    - The closer to zero, the weaker the relationship.
  - Positive values mark a positive correlation, while negative values indicate a negative correlation.



$$r = \frac{\sum\left[(x_i - \overline{x})\,(y_i - \overline{y})\right]}{\sqrt{\underbrace{\Sigma(x_i - \overline{x})^2}_{\text{Sum of the squared deviations for X}} * \underbrace{\Sigma(y_i - \overline{y})^2}_{\text{Sum of the squared deviations for Y}}}}$$

Evaluation Metrics (Cliff’s Delta)

$$\delta(i,j) = \begin{cases} +1, & x_i > y_j \\ -1, & x_i < y_j \\ 0, & x_i = y_j \end{cases}$$

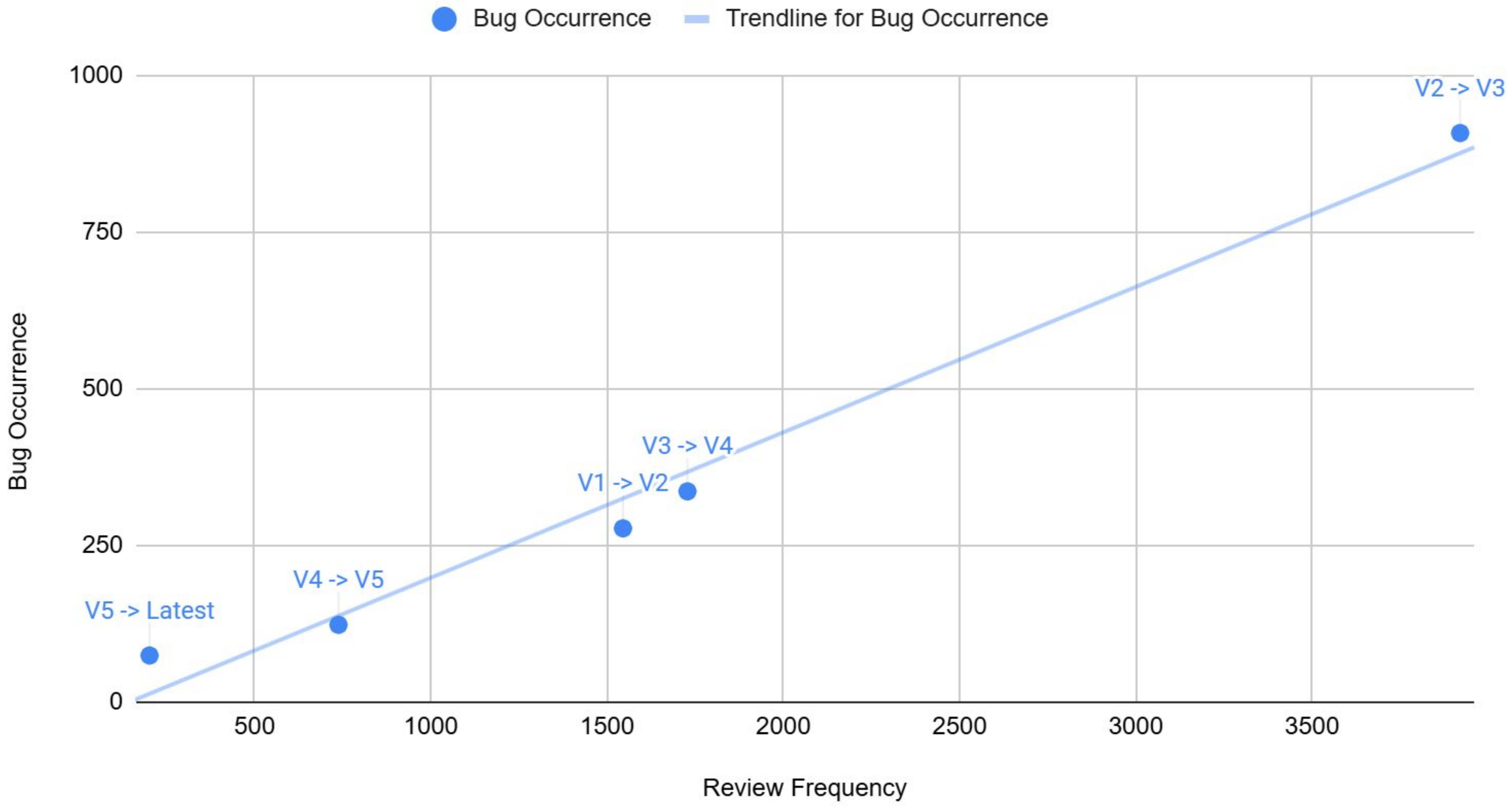
$$\delta = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\delta(i,j)$$

- Cliff’s δ
  - An effect size metric meant to compare two samples of ordinal data.
  - Essentially tracks the tendency of one value to being larger than another.
  - -1 ≤ δ ≤ 1
    - As the delta gets near ±1, it indicates a larger **absence** of overlap between the two samples, while a value closer to 0 indicates a greater degree of overlap.

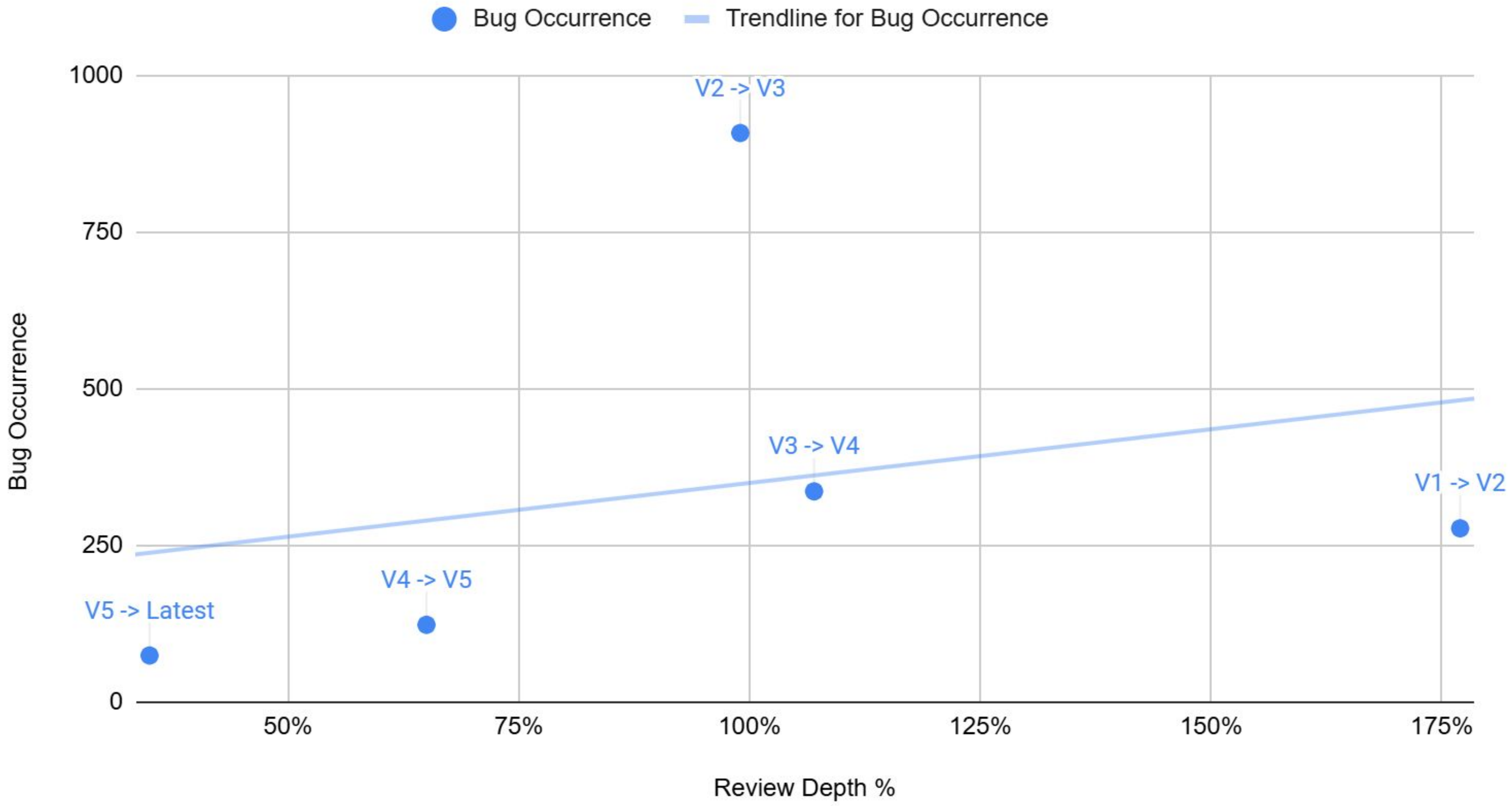
Results

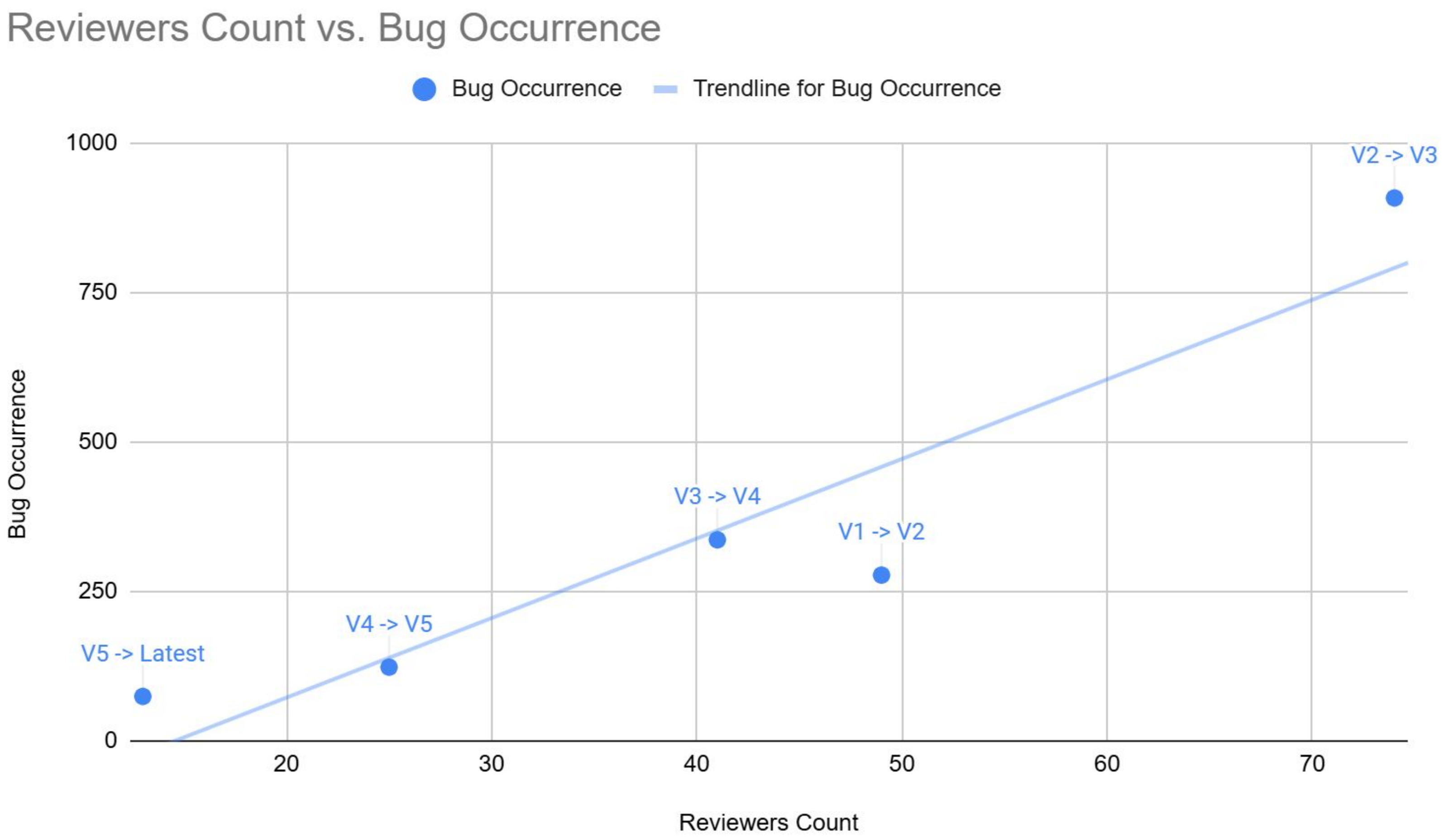
Release Version	V1 -> V2	V2 -> V3	V3 -> V4	V4 -> V5	V5 -> V5.04
Bug Occurrences	278	909	337	124	75
Review Frequency	1545	3920	1728	738	202
Review Depth	177%	99%	107%	65%	35%
Reviewer Count	49	74	41	25	13

Review Frequency vs. Bug Occurrence



Review Depth (%) vs. Bug Occurrence





Reviewers Count vs. Bug Occurrence
● Bug Occurrence
Trendline for Bug Occurrence
1000
750
500
250
0
20
30
40
50
60
70
Bug Occurrence
Reviewers Count
V2 -> V3
V3 -> V4
V1 -> V2
V4 -> V5
V5 -> Latest

Evaluation Results

- Correlation Coefficient
  - Review Frequency vs. Bug Occurrence: r = 0.996
  - Review Depth vs. Bug Occurrence: r = 0.758
  - Reviewer Count vs. Bug Occurrence: r = 0.953
- Cliff’s Delta
  - Review Frequency vs. Bug Occurrence: 0.68
    - This suggests a moderate-large effect size.
  - The delta of review depth and reviewer count show an incredibly large effect-size, but due to the nature of these parameters, it is largely irrelevant.
    - Review Depth vs. Bug Occurrence: -1.0
    - Reviewer Count vs. Bug Occurrence: -1.0

Interpretation

- There is a large positive correlation between both the frequency of reviews and bug occurrences, as well as the amount of individual reviewers and the number of bugs.
    - Review depth also has a fairly large positive correlation with bugs.
- None of these things necessarily mean that a larger amount of longer reviews with more reviewers equates to more bugs.
    - The high correlation could likely reflect the fact that buggier code requires a larger number of detailed code reviews by more reviewers, as more fixes need to be checked.
- Cliff's Delta shows that the amount of code reviews will always have a large degree of overlap with the amount of bug occurrences.
    - This can be interpreted much the same as the correlation, showcasing the need for more reviews with buggier code
    - The delta with regards to review depth and reviewer count shows overwhelming overlap, but can be thrown out as irrelevant, due to the fact that those values would never be larger than the amount of bugs.
        - One tracks the level of detail in a releases' reviews as a ratio to the overall level of detail, while the other only tells of the number of hands working on reviews.

Reflection

- This project taught the nuances of interpreting statistics, especially in a computer science research context.
  - Understanding specific formulas, like the correlation coefficient and cliff's delta, while also trying to correctly input our data into them.
  - A need for further research into the applicability of testing statistics (Z-Test, T-Test, Chi-Square Test) in a dataset with no random samples.
- We self taught on how to use and interact with the GitHub API as a means to fetch data.
  - There was a limitation to how much API data could be fetched within a certain timeframe.
- The lack of a strong conclusion highlights the need of a qualitative component. Something that could examine the logic behind the high correlation.
  - Possibly comparing the differences of reviews in an individual release.
  - Roadblocks to that approach included time and a lack of access to information.

GitHub Repository

https://github.com/RhoAI/CS563-Study-Project