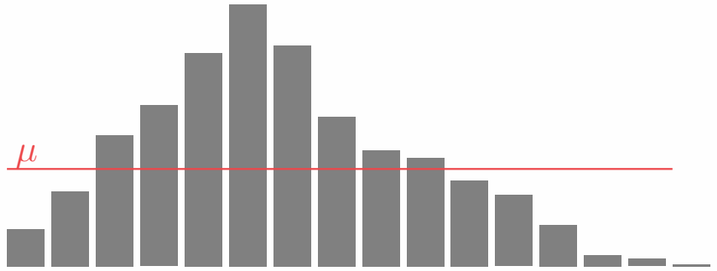
Source Code Volatility

Yegor Bugayenko

Lecture #12 out of 24
80 minutes

Volatility Metric



"The variance $Var(g)$ is the **Volatility** of the source code. The smaller the Volatility the more *cohesive* is the repository and the smaller the amount of the abandoned code inside it."

Then, the mean μ is calculated as:

$$\mu = \frac{1}{Z}\sum_{j=1}^{Z} g_j \tag{5}$$

Finally, the variance is calculated as:

$$\mathrm{Var}(g) = \frac{1}{Z}\sum_{j=1}^{Z} |g_j - \mu|^2 \tag{6}$$

The variance $\mathrm{Var}(g)$ is the Volatility of the source code.

Volatility vs. Number of Files in a Repo

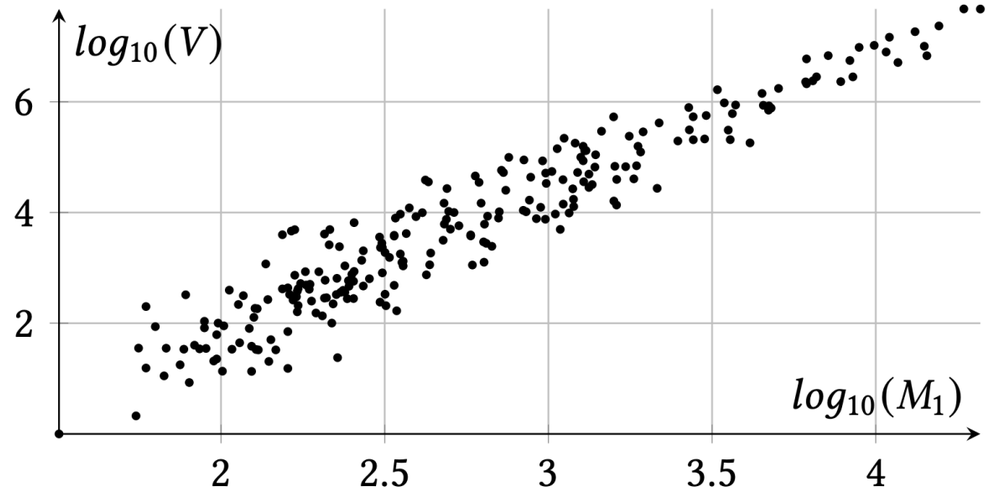


Figure 3. The relationship between the number of files in the repository and its Volatility (Z is set to 32)

Monolithic Repositories

Centralization The codebase is contained in a single repo encompassing multiple projects.

Visibility Code is viewable and searchable by all engineers in the organization.

Synchronization: The development process is trunk-based; engineers commit to the head of the repo.

Completeness Any project in the repo can be built only from dependencies also checked into the repo. Dependencies are unversioned; projects must use whatever version of their dependency is at the repo head.

Standardization A shared set of tooling governs how engineers interact with the code, including building, testing, browsing, and reviewing code.

Source: *Advantages and Disadvantages of a Monolithic Repository: A case study at Google*, Ciera Jaspan et al., ICSE, 2018

+2pt

“Our survey results show that engineers at Google strongly prefer our monolithic repo, and that visibility of the codebase and simple dependency management were the primary factors for this preference.”

— *Advantages and Disadvantages of a Monolithic Repository: A case study at Google*, Ciera Jaspan, Matthew Jorde, Andrea Knight, Caitlin Sadowski, Edward K. Smith, Collin Winter, Emerson Murphy-Hill, ICSE, 2018

+2pt

"At Google, almost all code exists in a single large, central repo, in which almost all code is visible to almost all engineers. The repo is used by over 20,000 engineers and contains over 2 billion lines of code."

— *Advantages and Disadvantages of a Monolithic Repository: A case study at Google*, Ciera Jaspan, Matthew Jorde, Andrea Knight, Caitlin Sadowski, Edward K. Smith, Collin Winter, Emerson Murphy-Hill, ICSE, 2018

+2pt

“Facebook’s main source repository is enormous–many times larger than even the Linux kernel, which checked in at 17 million lines of code and 44,000 files in 2013.”

— *Scaling Mercurial at Facebook*, Durham Goode et al., 2014

+2pt

“Before monorepo, I had to upgrade every package manually, which resulted in dissonance: one package used Symfony\Console 3.2, but other only 2.8 and it got messy for no reason.”

— *How Monolithic Repository in Open Source saved my Laziness*, Tomas Votruba, 2017

Benefits of “Manyrepo” Approach

Encapsulation Each repo encapsulates and hides its details from everybody else.

Fast Builds When a repo is small, the time its automated build takes is small.

Accurate Metrics Calculating LoC for a large repository doesn’t make any sense.

Homogeneous Tasks It’s easier to make tasks similar in size and complexity.

Single Coding Standard Smaller repositories look more beautiful.

Short Names Smaller namespaces mean better maintainability.

Simple Tests More dependencies are difficult to mock and test.

Source: Monolithic Repos Are Evil (2018)

Read this:

Volatility Metric to Detect Anomalies in Source Code Repositories, Yegor Bugayenko, Proceedings of the 1st ACM SIGPLAN International Workshop on Beyond Code: No Code, 2021

Advantages and Disadvantages of a Monolithic Repository: A case study at Google, Ciera Jaspan, Matthew Jorde, Andrea Knight, Caitlin Sadowski, Edward K. Smith, Collin Winter, Emerson Murphy-Hill, Proceedings of the International Conference on Software Engineering, 2018

Monolithic Repos Are Evil (2018)