MEASUREMENT REPORT · CHROMIUM 2024

CodeDelta scanned Chromium! — a massive 43.5 million lines of code

One calendar year of development on the codebase behind the world's browsers, measured statement by statement in a single command: 4.53 million logical statements of churn across a 326,948-file tree, its full composition, and how to reproduce every figure.

First release 13 July 2026

ABSTRACT

Chromium is the open-source browser codebase behind Chrome, Edge, Brave, Opera and others — among the largest publicly accessible codebases in existence. We measured one calendar year of its development (snapshots at 31 December 2023 and 31 December 2024) with **CodeDelta** (codedelta.app) in its git-native mode: one command, both 43.5-million-line snapshots, full CHG/DEL/ADD classification of every logical statement. The year churned **4,534,768 logical statements** — 12,390 per day — and touched **143,396 files, 44% of the entire tree**. Of all that change, **9.1% was editing of statements that already existed (REWORK)**; the remainder was insertion and removal. That places Chromium squarely in the mega-project band established by our 1,412-measurement corpus (WebKit, Linux kernel, FreeBSD, IntelliJ: 5–10% REWORK), at 3.2× the Linux kernel's 2024 churn volume. The scan completed in 105 minutes on consumer hardware — a laptop with 8 GB of memory — with no sampling and no machine learning: deterministic arithmetic, reproducible from the two commit hashes in §05.

IN PLAIN TERMS

Ask CodeDelta

> We took two year-apart snapshots of the code that runs most of the world's web browsers and compared every statement. In one year its developers added 2.4 million statements, removed 1.7 million, and edited 411 thousand in place — touching nearly half the files in the project. The striking part is the shape: only about one change in eleven was an *edit* to existing code. Big, healthy projects at this scale all look like this — our measurements of WebKit, Linux, FreeBSD and IntelliJ show the same pattern — and knowing that pattern is what lets unusual development processes stand out.

01 · The codebase

All figures are CodeDelta's own measurements of the 31 December 2024 snapshot (new side of the window):

QUANTITY	MEASURED
Total lines (LOC)	43,491,657
Source lines (SLOC)	32,171,209
Logical statements (LLOC)	16,764,119
Files	326,948
Comment lines	5,758,295
Source volume	1.82 GB

> *For scale: 24× the largest subject in our nine-codebase study (PostgreSQL, 1.34M SLOC). LOC counts every line; SLOC excludes blanks and comments; LLOC counts logical statements — the unit all churn figures below use.*

02 · Method and instrumentation

The measurement is CodeDelta's standard git-native scan: `codedelta --git <old>..<new>` extracts both snapshots and computes the full churn classification — CHG (changed in place), DEL (deleted), ADD (added) — per logical statement, per file, per language, using the two-pass sequence-alignment method of Paper A. No sampling, no inference; the arithmetic is deterministic and re-runnable by anyone from public history.

The window is calendar 2024: the last commit of 2023 (`502dd460` , 31 Dec 2023 23:55 UTC) to the last commit of 2024 (`2fc24fd0` , 31 Dec 2024 15:57 PST) on `main` of `chromium.googlesource.com/chromium/src` . History was fetched shallow from June 2023 (25 GB pack, 7.8 million git objects, ~550,000 commits).

Hardware: a consumer laptop with 8 GB of memory, the repository and all working data on an external USB hard drive. Engine 1.8.6. Wall-clock for the full scan including report generation: **about 1 hour 50 minutes** — consistent within 3% across four independent runs.

03 · Results — what one year did

CHURN QUANTITY	LLOC	SHARE OF CHURN
Added (ADD)	2,395,495	52.8%
Deleted (DEL)	1,728,106	38.1%
Edited in place (CHG)	411,167	9.1%
Total churn (CRN)	**4,534,768**	100%

REWORK = 9.1% (one statement in eleven edited in place); REP_CHURN 0.909. Decomposed by change shape: REWRITE (co-located delete+add pairs) 149,543 · PURE_DEL 1,578,563 · PURE_ADD 2,245,952. Comment churn: 507,853 lines. Physical-line churn: 9,938,827 SLOC.

FILE ACTIVITY	FILES
Modified	83,187
Created	39,312
Deleted	20,897
Touched (44% of tree)	**143,396**
Unchanged	183,552

> *Rates, for intuition: 12,390 logical statements of churn per calendar day, weekends included; 27,155 physical source lines per day.*

04 · Reading the number

Chromium's REWORK of 9.1% lands inside the band our measurement corpus (1,412 repo-year measurements, 135 repositories, 2010–July 2026) establishes for mega-projects — same-method 2024 windows:

CODEBASE	WINDOW	CHURN (CRN LLOC)	REWORK
Chromium	2024	**4,534,768**	**9.1%**
WebKit	2024	4,973,778	6.0%
Linux kernel	2024	1,436,681	9.3%
FreeBSD	2024	955,664	8.2%

The pattern from Paper C holds at every scale we can find: mid-size established projects rework ~17% of their churn; the very largest projects run 5–10% — more growth surface, more vendored and generated content, more contributors landing new code — and intensively AI-agent-built codebases collapse below 1%. Chromium, at three times the kernel's churn, sits exactly where the stratum predicts. The measurements are what they are; readers are invited to draw their own conclusions and re-run the window.

05 · Limits and reproducibility

- **One window, one project.** This report is a scale demonstration and a corpus data point, not a study; the study is Paper C and the forthcoming baseline work.
- **Calendar windows include everything** — vendored rolls, generated files, mass renames. That is deliberate: the corpus measures development as it lands, and the composition tables let a reader separate growth from editing.
- **Reproduction:** `codedelta --git 502dd46098ac722e61912aa7b91a3ee370aa0e64..2fc24fd052f0bd7c692ddf375773ef2a58247701` against `chromium.googlesource.com/chromium/src`. The run record, with full provenance, is retained in the study's canonical results log.

References

1. Paper A (this series), *Automated Source Code Churn Measurement* — paper-churn.html
2. Paper C (this series), *REP_CHURN: Replacement-Dominant Churn as a Process Signature of AI-Assisted Development* — paper-repchurn.html
3. The Chromium Projects — chromium.org; source: chromium.googlesource.com/chromium/src