



Develocity — Product Description

Build Scan and Develocity Analytics

A Build Scan is a permanent, shareable record of exactly what happened in a single build or test run. It captures the build from the inside — task and goal timing, test results, how dependencies were resolved, which cache entries were hit or missed, the full environment the build ran in, and the precise cause of any failure — and publishes it as a URL that any engineer can open. Instead of scrolling through thousands of lines of CI log, a developer looks at a structured picture of the build and goes straight to what matters. Because a scan is just a link, it becomes the natural thing to paste into a pull request, a chat channel, or an incident ticket, which changes how teams collaborate on build and test problems. A particularly powerful feature is the ability to compare two builds side by side and see exactly what differed between them — a common way to answer "why did this pass yesterday and fail today?"

Develocity Analytics sits on top of the accumulated history of every Build Scan and turns it into organization-wide insight. Platform and engineering leaders can see trends over time and across teams: which projects are slow, which tests fail most often, how effectively the cache is working, and where build time and compute spend are actually going. This moves decisions about developer productivity from anecdote to evidence — you can identify the real bottlenecks, act on them, and then measure whether the fix worked.

Build Cache

The Build Cache eliminates redundant work. Every build task produces an output — compiled code, generated sources, test results — and the cache stores that output keyed on the task's inputs. Whenever the same work is requested again, and nothing relevant has changed, the result is fetched in seconds instead of being recomputed. This works locally on each developer's machine and, more importantly, as a shared remote cache across the whole organization: if one developer or CI agent has already built or tested something, everyone else reuses that result rather than repeating it.

In a large codebase, most build and test activity is repetition — the same modules are rebuilt and the same tests re-run over and over, on laptops and across CI, with only small parts actually changing. The Build Cache removes that waste without changing how anyone works; the same commands run, they simply finish far sooner. The result is faster feedback for developers, shorter CI queues, and lower infrastructure cost, because the same hardware does dramatically less repeated computation. Crucially, the cache spans developer machines and every CI system in use, so the savings apply across the entire estate rather than being trapped inside a single CI vendor's per-pipeline cache.



Maintenance and Support

A Develocity subscription includes ongoing maintenance and support directly from Gradle. The full terms are set out in the Gradle Software License Agreement:

<https://develocity.ai/legal/gradle-software-license-agreement/>

Product and upgrade support. Gradle provides assistance with installation, configuration, performance tuning, day-to-day operation, and troubleshooting of supported releases, as well as hands-on help upgrading from any version — supported or not — to a current one. Support is delivered through a priority support channel provided on purchase, with an unlimited number of support requests, during business hours (12 hours per day, 06:00–18:00 Eastern Time or 06:00–18:00 London time, Monday to Friday, excluding published holidays).

Support service levels. Gradle works to the following response-time targets:

Severity	Initial response	Workaround target	Resolution target
1 — Critical	Within 4 hours	1 business day	Next patch release
2 — Major	Within 4 hours	5 business days	A future patch release
3 — Minor	By close of next business day	Case by case	A future major or patch release
4 — Minimal	By close of next business day	Case by case	A future major or patch release

Maintenance updates. Non-breaking improvements, dependency updates, and minor fixes are delivered continuously as patch releases, along with access to updates as they become available.

Release lifecycle. Each Develocity release is supported for 18 months from its general-availability date, with clear published release and end-of-life dates so upgrades can be planned well in advance.

Security updates. Security vulnerabilities — including those in embedded third-party libraries — are patched to defined timelines based on severity. Critical issues are mitigated within 7 days and resolved within 30, covering the current and two previous major releases; high-severity issues are addressed within 30 days on the current release. Customers are notified through release notes, advisories, and a security mailing list, and all first-party CVEs are published at security.gradle.com.