



Build times breaking your flow?

Learn how to make them fast!

Droidcon London,
October 2025

A few build questions?

- ◆ Who has a long build?
 - 1 hour and more?
 - 30 minutes and more?
 - 5 minutes and more?
 - Less than a minute?
- ◆ Who has a complex build?
- ◆ Who is happy to use a build tool?



Agenda

Introduction

Gradle & Android Gradle Plugin

Concepts & Definitions

Gradle Scalability

What's next?

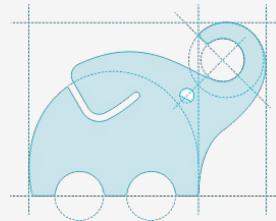


Introduction

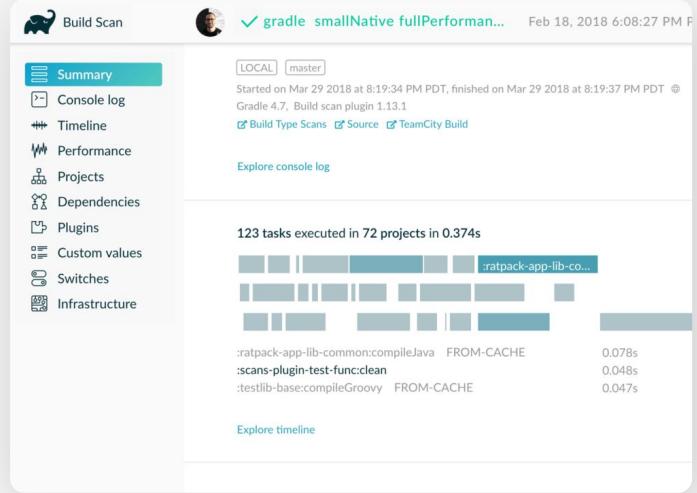




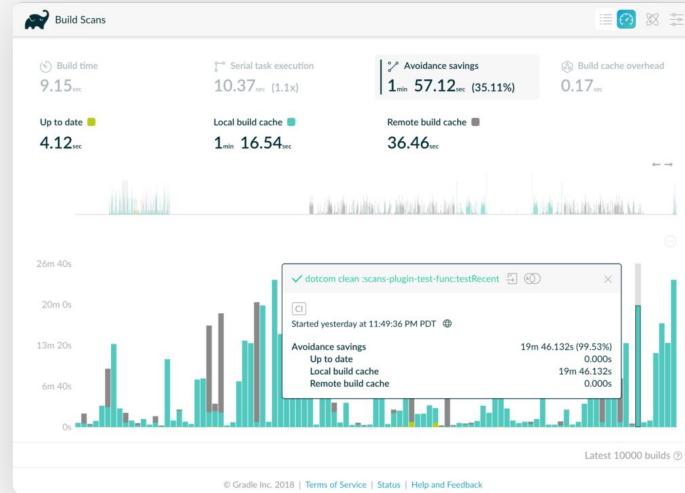
Since 2008, our mission is
to improve
developer productivity



Why Developcity? It's about the data.



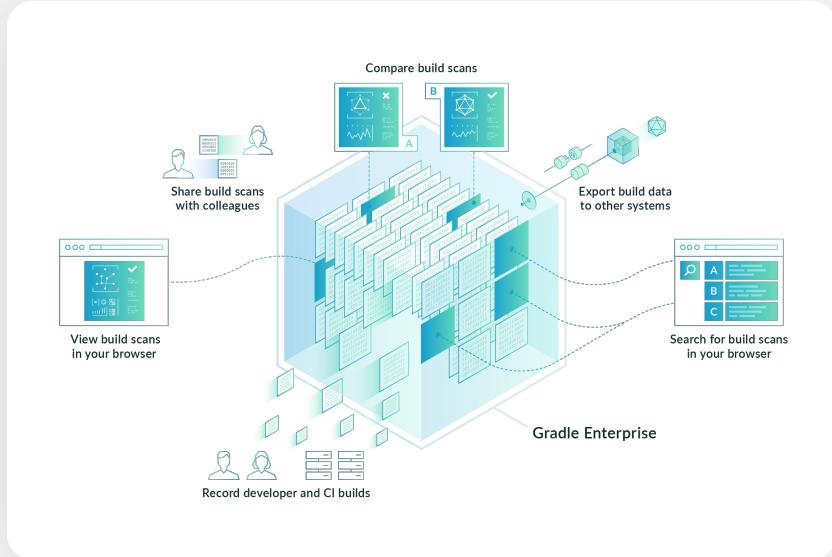
Build Scan - task-level analytics for a single build execution



Dashboard - analytics and insights across last 10,000 local and CI builds



Developity is a data platform



- ◆ **Collective team data** - all the data about every build across the team creates unique dataset and insights.
- ◆ **Build Performance Management** - only with collective, deep data will builds stay fast and reliable.
- ◆ **Debugging acceleration** - only with collective deep data is it possible to quickly discover the root cause for build failures.



Gradle & Android plugin



Gradle Vision

Gradle Build Tool is an open-source build system that automates the process of building software of any type, size, or complexity in a fast and reliable manner.



Gradle Vision

Gradle Build Tool is an open-source build system that automates the process of **building software of any type, size, or complexity in a fast and reliable manner.**



Gradle

01

Configuration and execution engine

02

APIs allowing to build abstractions for common development tasks

03

Plugins that enables developer to describe their build, leaving Gradle to orchestrate its execution



Java application sample

```
plugins {
    id("application")
}

repositories {
    mavenCentral()
}

dependencies {
    testImplementation("org.junit.jupiter:junit-jupiter-engine:5.9.3")
    testRuntimeOnly("org.junit.platform:junit-platform-launcher")
    implementation("com.google.guava:guava:32.1.1-jre")
}

application {
    mainClass = "com.example.Main"
}
```



Android Gradle Plugin

01

Developed by Google, on top of Gradle APIs

02

Defines models and abstractions for Android development

03

Since version 7.0, the major version number is aligned on the Gradle one



Android example

```
plugins {
    id("com.android.application") version "8.9.1"
}

android {
    namespace = "com.example.app"
    compileSdk = 34
    // ...

    defaultConfig {
        applicationId = "com.example.app"
        minSdk = 34
        // ...
    }
}
```

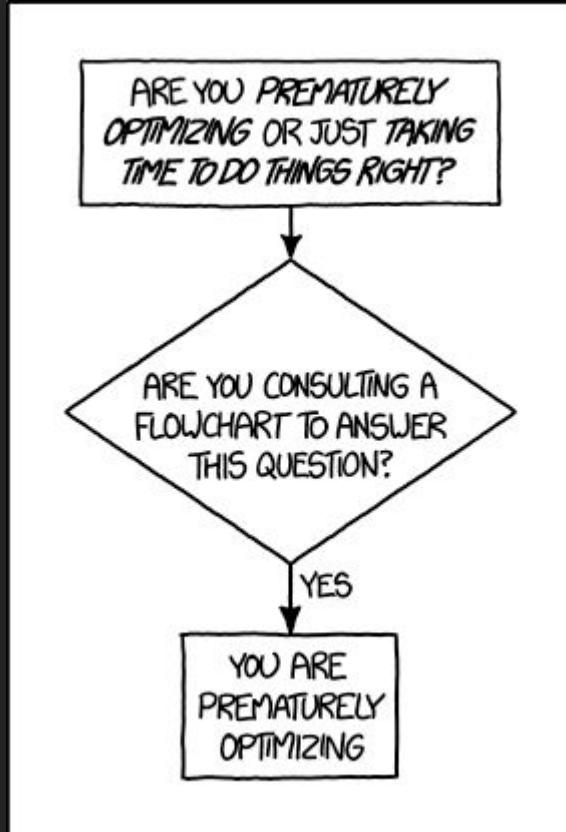


Concepts & Definitions



To measure is to doubt

- ◆ Measure
- ◆ Change
- ◆ Doubt
- ◆ Measure
- ◆ Compare
- ◆ Repeat



xkcd.com/1691



Development loops

01

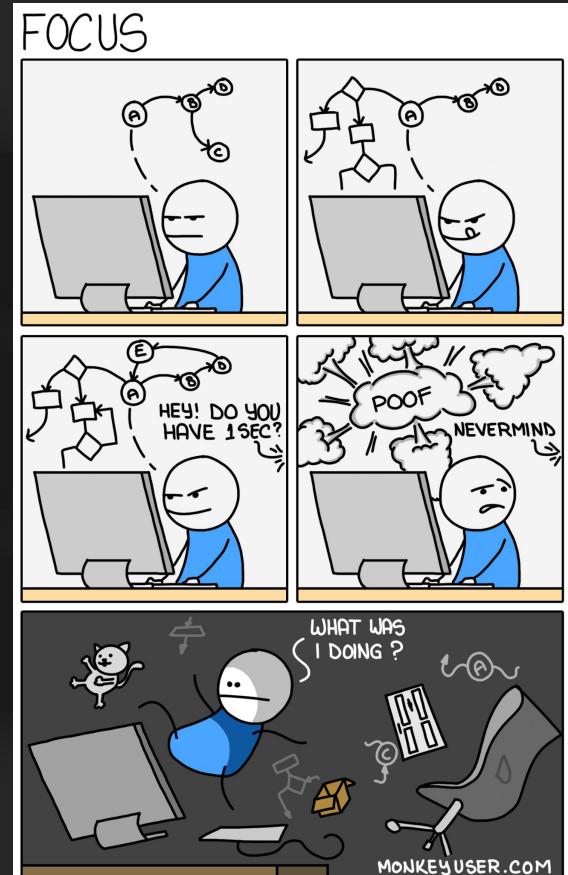
Internal development loop
Code, Test, Deploy, ...

02

External development loop
IDE sync, CI stateful or ephemeral, ...

03

Build development loop
Project configuration, changing build logic, ...



Guiding principles

01

Overhead must be proportional to the size of the change

02

Developers must have a comprehensive understanding of the build



Main axes

01

Work avoidance

Do not redo what's already been done.

02

Parallelism

Maximise speed by leveraging parallel execution.



Gradle Scalability



Gradle Scalability

In practice



Work avoidance

Already available

	Dev	Build logic/Sync	CI	Ephemeral CI
Dependency Cache	✓	✓	✓	⚠️
Configuration cache hit	✓	✓ / N/A	✓	✗*
Up-to-date task?	✓	✓	✓	✗
Compilation avoidance (ABI)	✓	✓	✓	✗



Work avoidance

Already available

	Dev	Build logic/Sync	CI	Ephemeral CI
Local Build Cache	✓	✓	✓	🏗️
Remote Build Cache (Develocity)	✓	✓	✓	✓



Work avoidance

Already available

	Dev	Build logic/Sync	CI	Ephemeral CI
Configure on Demand (Incubating)	⚠	⚠	⚠	⚠
Incremental tasks	✓	✓	✓	✗
Predictive Test Selection (Develocity)	✓	✓ / N/A	✓	✓



Work avoidance

Under development

- ◆ Isolated Projects
 - Incremental per-project Configuration Cache for task execution
 - Incremental per-project Cache of tooling models for Android Studio sync



Parallelism

Already available

	Dev	Build logic/Sync	CI	Ephemeral CI
Parallel CC load (8.11)	✓	✓ / N/A	✓	?
Parallel CC write (Incubating 8.11) i	✓	✓ / N/A	✓	?
Intra-project parallelism through CC	✓	✓ / N/A	✓	?
Parallelism with <code>--parallel</code> i	✓	✓	✓	✓



Parallelism

Already available

	Dev	Build logic/Sync	CI	Ephemeral CI
Parallel downloads	✓	✓	✓	✓
Parallel execution inside a task (Worker API)	✓	✓	✓	✓
Test Distribution (Develocity)	✓	✓ / N/A	✓	✓
Parallel model building in Android Studio	N/A	N/A / ✓	N/A	N/A



Parallelism

Under development

- ◆ Isolated Projects
 - Parallel project configuration for task execution
 - Parallel project configuration for Android Studio sync



Setting expectations ...

Initialization

Configuration

*build
models*



:proj-a ... :proj-z

~ 30%

~ 60%



What's next?



Forward Looking Statements Disclaimer

This document may contain forward-looking information and statements about Gradle, its subsidiaries, and their products. These statements include projections and estimates, statements regarding plans, objectives and expectations with respect to future operations, products and services, and statements regarding future performance. Forward-looking statements may be identified by the words “believe”, “expect”, “anticipate”, “target” or similar expressions. Although Gradle's management believes that the expectations reflected in such forward-looking statements are reasonable, you are cautioned that forward looking information and statements are subject to numerous risks and uncertainties, many of which are difficult to predict and generally beyond the control of Gradle, that could cause actual results and developments to differ materially and adversely from those expressed in, or implied or projected by, the forward-looking information and statements. Gradle undertakes no obligation to publicly update its forward-looking statements, whether as a result of new information, future events, or otherwise.



Gradle 9

9.0.0 released on July 31st 2025, 9.1.0 released on September 18 2025,

9.2.0 released on October 29 2025

- ◆ Kotlin 2.1 & Language Version 2.1
 - Stricter nullability with JSpecify
- ◆ Groovy 4
- ◆ Java ≥ 17 for Gradle execution 
 - Worker, compilation and tests still support Java ≥ 8
- ◆ Configuration cache: No more errors when using core Gradle plugins
 - But some still disable CC as they remain incompatible
- ◆ 2 years and more of API cleanups 
- ◆ Performance improvements and memory usage reduction with 9.2.0



Gradle 9.x and after

Actively worked on

- ◆ Declarative Gradle
 - Composition and extensibility research
 - Plan for productization
- ◆ Resilient IDE sync
 - Recover from errors on upgrade or invalid changes
- ◆ Continued maintenance and evolution
 - Performance improvements
 - Memory usage reduction
 - Bug fixing

Gradle 9.x and after

Work avoidance

- ◆ Deprecate disabling Configuration Cache
- ◆ New generation build cache
- ◆ Coarse grained caching (task subgraph)
- ◆ Reusable worker daemons
- ◆ Focus



Gradle 9.x and after

Parallelism

- ◆ Streaming task execution (via CC)
- ◆ General contention improvements (reduce locking)



Public Gradle Build Tool Roadmap

<https://roadmap.gradle.org>



Thank you!



```
speaker {  
    name = "Louis Jacomet"  
    twitter = "@ljacomet"  
    mastodon = "@ljacomet@foojay.social"  
    github = "ljacomet"  
}
```



Questions ?

