**Research Document**

**P-CB-S304**

**PC Parts Web Shopping**



**Name:** Hristo Hristov

**Student Number:**

**Email:** [h.hristov-ab@student.fontys.nl](mailto:h.hristov-ab@student.fontys.nl)

Jacco Snoeren – [j.snoeren@fontys.nl](mailto:j.snoeren@fontys.nl)

Onno Marsman – [o.marsman@fontys.nl](mailto:o.marsman@fontys.nl)

**Fontys University of Applied Sciences**

**Eindhoven – Netherland**

**2020-2021**

Table of Contents

[What are the differences between the most popular java build tools? 3](#_Toc87045123)

[Which are the most popular java build tools? 3](#_Toc87045124)

[How do they compare? 3](#_Toc87045125)

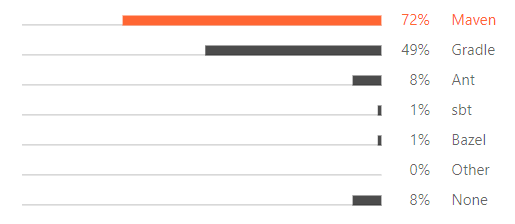
[Which one should you choose a person chose if starting a new project? 5](#_Toc87045126)

[Conclusion 5](#_Toc87045127)

# What are the differences between the most popular java build tools?

## Which are the most popular java build tools?

Based on research, the most popular build tools amongst developers are Maven and Gradle.



The difference between Maven and Gradle is flexibility, performance, user experience, and dependency management.

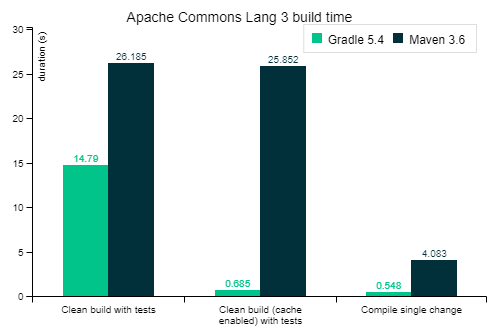
## How do they compare?

* **Flexibility**

Gradle is modeled in a way that is extensible in the most fundamental ways. Gradle's model also allows it to be used for native development with C/C++ and can be expanded to cover any ecosystem. Maven provides a very rigid model that makes customization tedious and sometimes impossible. Any Maven build can be easy to understand, as long as you don’t have any special requirements, but this also makes it unsuitable for many automation problems. Gradle, on the other hand, is built with an empowered and responsible user in mind.

* **Performance**

This is the comparison between Gradle and Maven from the performance side.

****

Gradle has 3 features that make its performance much better than Maven:

* **Incrementality,** Gradle only runs things that are necessary and only files that are changed when it’s possible. It doesn’t track input and output of task.
* **Build Cache,** Gradle will reuse the build output of any other Gradle build with the same input including between the machines.
* **Gradle Daemon**, Gradle keep the build information “hot” in memory.
* **User Experience**

The fact that Maven has been around for a longer time means that its support through Integrated Development Environments is better for many users. Gradle's IDE support continues to improve quickly, however.

Although IDEs are important, a large number of users prefer to execute build operations through a command-line interface. Gradle provides a modern CLI.

Finally, Gradle provides an interactive web-based UI for debugging and optimizing builds.

* **Dependency Management**

As a library customer, Maven allows one to override a dependency (only by version), and for Gradle, it provides a customizable dependency selection and substitution rules, it can be declared once and being handle unwanted dependencies project-wide.

Maven works with the shortest path on the dependency conflict resolution, it’s impacted by declaration ordering. For Gradle, it selects the highest version of a dependency found in the graph because of conflict resolution.

Maven has a few built-in dependency scopes, which means there’s no separation between unit and integration test. Gradle allows custom dependency scopes, which provides better modeled and faster builds.

## Which one should you choose a person chose if starting a new project?

What you choose will depend primarily on what you need. Gradle is more powerful, however do you really need all features and functionalities it offers. Maven might be best for small projects, while Gradle is better for bigger projects. Even if a project is started using one of the build tools, the project can be migrated to the other tool.

# Conclusion

In conclusion I personally choose Gradle as my Java build tool, because even though I will use it for a project of a smaller scope, later down the road it will probably be the main build tool I will use when working in the industry.