How a $7B Bank Saved Money, Reduced CI Times By 62% and Improved Developer Productivity

Banking & Finance

$7.2B Revenue

Europe

Our Nx Enterprise customer, a $7B European-based financial services company, had previously adopted an Nx monorepo in order to improve code shareability, reduce duplicate code, and simplify their dependency management. They had recently incorporated an on-premise deployment of Nx Cloud to take advantage of Nx Replay and Nx Agents to further improve their CI times and, by extension, the team's velocity and product time to market.

Anticipating additions to their monorepo in the coming year, they wanted to ensure the developer experience of the monorepo didn't suffer as it grew. Their ongoing partnership with us through Nx Enterprise allowed them to continuously enhance their performance as their workspace matured.

Problem

Having been Nx users for several years and recently deployed Nx Cloud, they had already seen some productivity gains but were experiencing slower than desired CI runs, impacting their ability to merge and release features quickly.

During our regular Nx Enterprise check-ins, we identified potential misconfiguration of Nx which may have caused their projects to be deployed unnecessarily upon any dependency change and may have resulted in longer test and lint runs. We also suspected their micro frontends were also sub-optimally split. It was clear that a combined working session would result in substantial productivity gains.

Solution

The Nx team partnered with the customer over two days to complete a comprehensive review of their configuration and architecture. The session allowed us to understand their setup, organizational goals, and co-develop guidelines for improvements.

During the configuration review we were able to identify and remove or replace several unused computation-heavy lint rules and provide guidance on how to strategically design project boundary tags to achieve better repository clustering and access control. The team was also able to help them improve their unit test setup to achieve shallow depth with significantly reduced transpilation cost, include additional task parallelization, and improve task inputs to more frequent cache hits on build and deploy tasks.

A review of their existing architecture allowed us to provide guidelines on how to make improvements to their micro frontend solution and how to create better separation and optimize affected runs.

During a follow up Q&A session we helped them clarify and understand Nx best practices, how to create repeatable processes, and identify steps to further enhance productivity going forward.

Team

1	5	2
Customer Dev Platform Team Lead	Customer Dev Platform Team Members	Nx Developer Productivity Engineers

After the improvements to their lint and test tasks, the customer was able to use fewer, less resource-intensive machines in CI, allowing them to reduce cost while increasing performance.

In addition to these results, we identified several tasks they could work on over the coming months to further improve their local and CI runs, helping to keep their team productive.

Conclusion

Nx can make significant improvements to any team's CI and developer experience; but for enterprise-scale engineering teams, on-going partnership through Nx Enterprise is imperative to scalability and developer productivity as their organization and workspaces grow.

Results

Overall CI Improvement	Average CI Time	Average Test	Average Lint
62%	19m from 50m+	8m40s from 35m+	1m20s from 5m+

"Last week, I had the pleasure to meet two Nx team members. It was a blast! We had a two-day session to improve our @NxDevTools usage.

Key numbers on repo with 480 projects: lint 1m 20s (from 5m+), test 8m 40s (from 35+), full CI down to 19m with same agents.

There are 20+ new tasks on the backlog as a result of this session. These will only provide further improvements for both local and CI runs.

I can't thank you enough!"

Dev Platform Team Lead