Software Release Check

Where did your last release spend its time, and what does that cost you a year?

Expeditious Software
contact@es.nl

Sample report · Platform team, 120 engineers

YOUR RESULT

Your release work costs about EUR 59,000 a year: 740 hands-on hours across 2 to 3 products.

Based on 60 releases a year at 6.5 hours each, including fixing failed releases, plus 345 upkeep hours, at EUR 80 an hour.

Your last release took about 3 working days. People spent about 5 hours on it. The rest was waiting.

Most of it sits in the process: waits at an approval or change board, 3 teams.

FIXES FOR YOUR SITUATION

1. Move approvals into the pipeline: automated checks on every change, and a human approval only where your policy requires one.
2. Remove the handoff: one pipeline the team starts itself, with status posted where everyone can see it.
3. Automate the manual steps, starting with the one that runs most often.

DORA SNAPSHOT

Use these as your current numbers, recheck monthly, and track the trend, not perfection.

Deployment frequency

Medium → Weekly to monthly

Developing deployment frequency: releases happen, but not yet as a flow.

Your current cadence suggests larger, less frequent releases. Focus on automating your deployment pipeline end-to-end, reducing manual checks, and encouraging smaller changes so you can move from weekly or monthly releases towards at least weekly, then daily deployments.

Lead time for changes

High → 1 day to 1 week

High lead time performance: changes usually reach production within a week.

To reach elite lead times, inspect where changes wait: in code review, in queues for testing, or in release approval. Automate test suites, streamline approvals, and encourage smaller pull requests so changes can flow from commit to production in hours rather than days.

Change failure rate

Medium → 2 to 3 of the last 10 releases

Developing stability: a noticeable share of deployments cause problems.

Your failure rate suggests gaps in testing, rollout strategy, or environment consistency. Prioritise stabilising test suites, improving pre production environments, and adopting safer deployment patterns like blue green or canary releases so you can reduce the percentage of changes that trigger incidents.

Time to restore service

High → 1 hour to 1 day

High recovery performance: incidents are usually resolved within the same day.

To reach elite recovery times, focus on faster detection and clearer first-response playbooks. Standardise alerts on user-impacting symptoms, ensure on-call engineers can roll back or disable problematic changes quickly, and reduce dependencies on manual coordination during incidents.

NEXT STEP

A **Release Bottleneck Review** measures this properly in two weeks and ends in a decision with its cost.

Book a conversation: book.es.nl/appointment/2