



Thank you, **Mateusz Ulas**. Below is your delivery performance summary based on your responses.

### Scorecard, current state

Use as baseline, recheck monthly, track trend not perfection.

#### Deployment frequency

**Medium → Once per week to once per month**

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

**High → 6-15%**

High stability - only a small fraction of deployments cause issues.

To push towards elite levels, focus on eliminating recurring patterns of failure. Strengthen automated tests around known weak spots, use feature flags or canary releases for risky changes, and run small post-incident reviews to capture and apply lessons quickly.

#### Time to restore service

**Medium → 1 day to 1 week**

Developing recovery - some incidents take many hours or days to resolve.

Introduce a basic incident management model: a clear on-call rota, a single incident channel, and simple runbooks for your most critical services. Strengthen monitoring so you can detect issues earlier, and practice rollback and failover so teams are not improvising under pressure.

### OVERALL

#### High performer

Overall, you are a **High performer** on delivery. You already release software with good speed and stability. The next step is to make this performance consistent across all teams and all services, and to invest in developer experience so that changes remain safe even as throughput grows.

### NEXT STEPS

If you want a deeper SDLC audit or help moving towards elite performance, send us an email at [contact@es.nl](mailto:contact@es.nl) or [schedule a call here](#).