Legacy
maaretp

# Making Releases Routine

Maaret Pyhäjärvi

https://www.linkedin.com/in/maaret/

@maaretp@mas.to

# 1ˢᵗ year: My Signature Move



30 days → 30 minutes
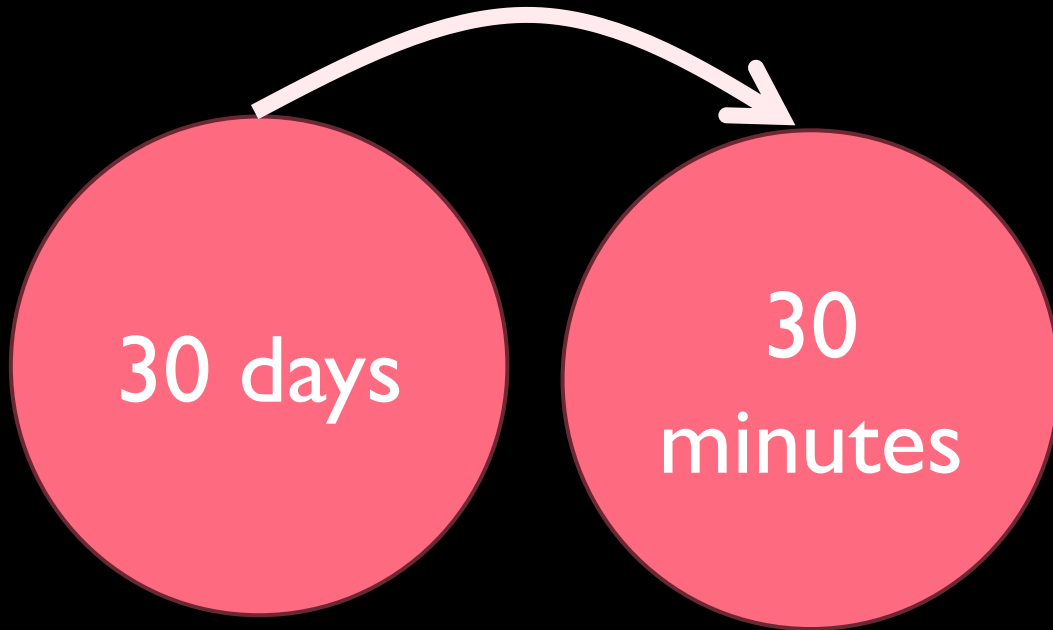
# Pool is not a bigger bathtub.

RELEASE FREQUENCY

MONTHS   WEEKS   DAILY   HOURLY   CONTINUOUS

TESTING {
- Separate test automation team
- Fast, comprehensive unit&functional tests as pipeline
- Devs&testers share test automation

ARCHITECTURE {
- Everything deployed together
- Services/components deployed independently

RELEASES {
- Release trains
- Decoupling deployment and release
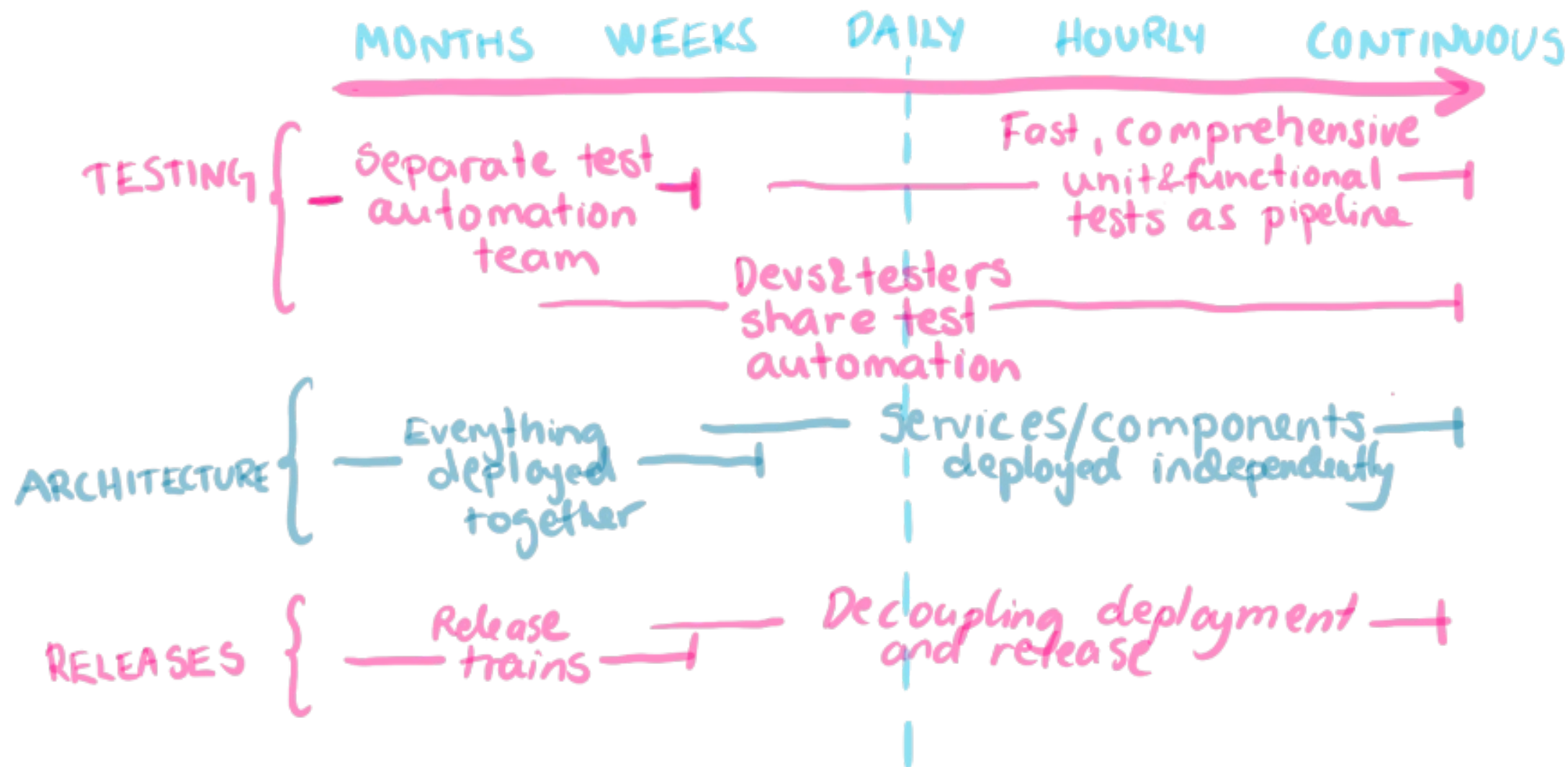
# 2ⁿᵈ year: Something Lost



30 minutes → 90 days
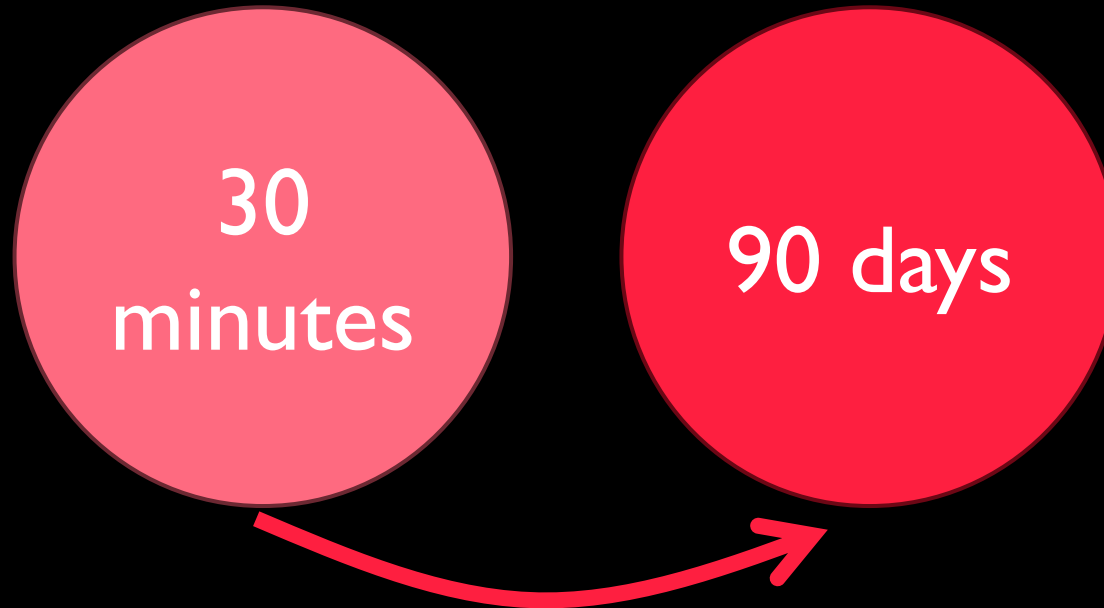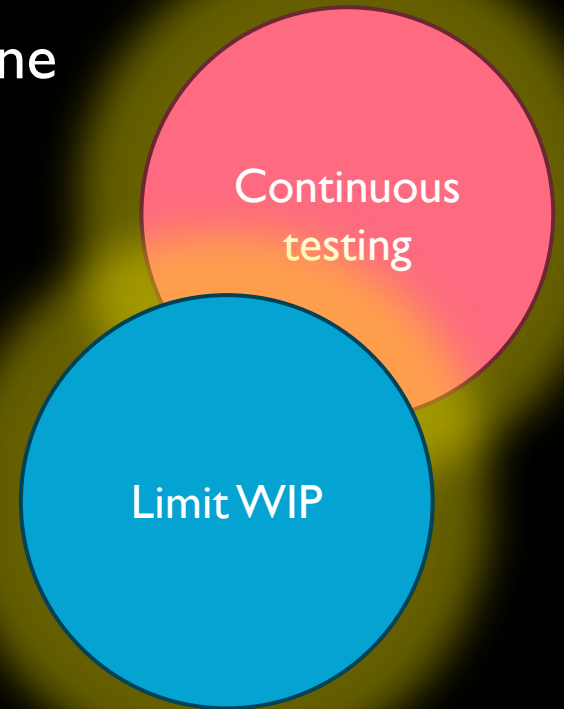
# The Failure

## Mistakes were made. And by me.

1. Making results gap visible is a testing skill. Just because someone tests, does not mean they test well.
   1. Recognizing scope
   2. Nudging scope with conversational approach
   3. Orchestrating fixes
   4. Feeling powerless
2. Premature done starts new work. New work piles up uncertainty.

Continuous testing

Limit WIP

# 3rd year: Make Releases Routine

90 days

30 minutes
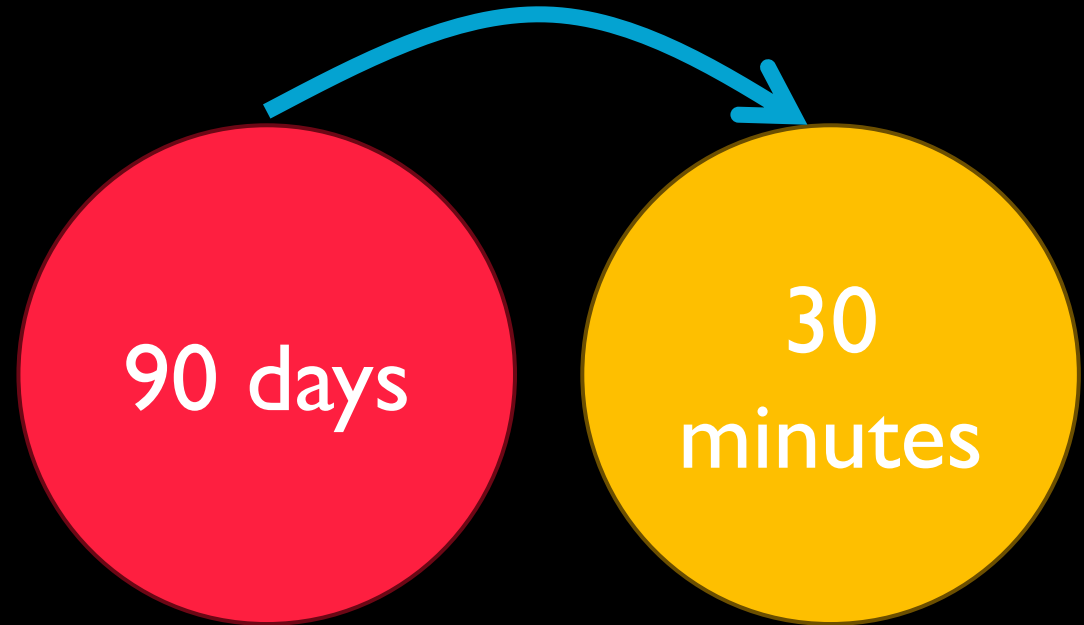
# Describing Release and Grouping Tasks



**TASKS**

**Write release notes** - 26 individual changes to message worth saying
**Create release checklist** - while I know it by heart, others may find it useful to tick off what needs doing to say its done
**Select / design title level tests** for test execution (evidence in addition to TA - test automation)
**Split epics** to this release - other release so that epics reflect completed scope over aspirational scope. and can be closed for the release
**Document per epic acceptance criteria**, esp. out of scope things - documentation is an output not input, but if I was testing, it was a daily output not something to catch up at release time
**Add Jira tasks into epics to match changes** - this is totally unnecessary but I do that to keep a manager at bay, close them routinely since you already tested them at pull request stage
**Link title level tests to epics** - again something normally done daily as testing progresses, but this time was left outside the daily routine
**Verify traceability matrix** of epics ('requirements') to tests ('evidence') shows right status
**Execute any tests in test execution** - optimally one we call release testing and would take 15 minutes on the staging environment
**Open Source license check** - run license tool, compare to accepted OSS licenses and update licenses.txt to be compliant with attribution style licenses
**Lock release version** - Select release commit hash and lock exact version with a pull request
**Review Artifactory Xray statistics** for docker image licenses and vulnerabilities
**Review TA (test automation) statistics** to see it's staying and growing
**Press Release-button in Jira** so that issues get tagged - or work around reasons why you couldn't do just that
**Run promotion that makes the release** and confirm the package
**Install to staging environment** - this is something from 3 minute run a pipeline to 30 minutes do it like a customer does it
**Announce the release** - letting others know is usually useful
**Change version for next release** in configs

**GROUP**

**START OF NEW**

Create release checklist
Change version for next release

**CONTINUOUS SYSTEM TESTING**

Select / design title level tests
Split epics
Document per epic acceptance criteria
Add Jira tasks into epics to match changes
Link title level tests to epics
Verify traceability matrix
Execute any tests in test execution

**SEPARATE CADENCE**

Review Artifactory Xray statistics
Review TA (test automation) statistics
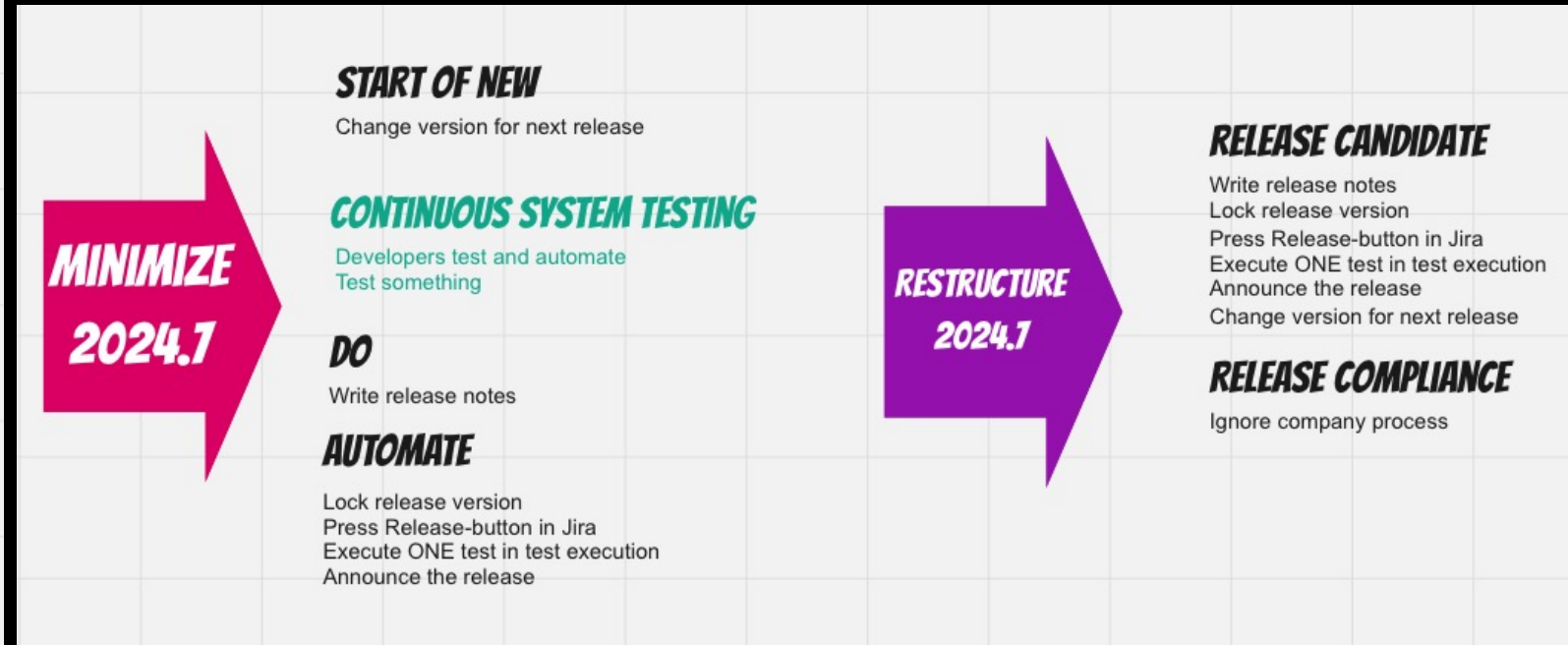Architecture review
Threat modeling

**DO**

Write release notes

**AUTOMATE**

Open Source license check
Lock release version
Press Release-button in Jira
Run promotion that makes the release
Install to staging environment
Execute ONE test in test execution
Announce the release

https://www.linkedin.com/in/maaret/    @maaretp@mas.to

# Experimenting with Structure



**FOCUS 2024.3**

**START OF NEW**
Create release checklist
Change version for next release

**CONTINUOUS SYSTEM TESTING**
Select / design title level tests
Split epics
Document per epic acceptance criteria
Add Jira tasks into epics to match changes
Link title level tests to epics
Verify traceability matrix
Execute any tests in test execution

**SEPARATE CADENCE**
Review Artifactory Xray statistics
Review TA (test automation) statistics
Architecture review
Threat modeling

**DO**
Write release notes

**AUTOMATE**
Run Open Source license check
Lock release version
Press Release-button in Jira
Run promotion that makes the release
Run Install to staging environment
Execute ONE test in test execution
Announce the release

**MINIMIZE 2024.7**

**START OF NEW**
Change version for next release

**CONTINUOUS SYSTEM TESTING**
Developers test and automate
Test something

**DO**
Write release notes

**AUTOMATE**
Lock release version
Press Release-button in Jira
Execute ONE test in test execution
Announce the release

**RESTRUCTURE 2024.7**

**RELEASE CANDIDATE**
Write release notes
Lock release version
Press Release-button in Jira
Execute ONE test in test execution
Announce the release
Change version for next release

**RELEASE COMPLIANCE**
Ignore company process

**Practice makes better!**

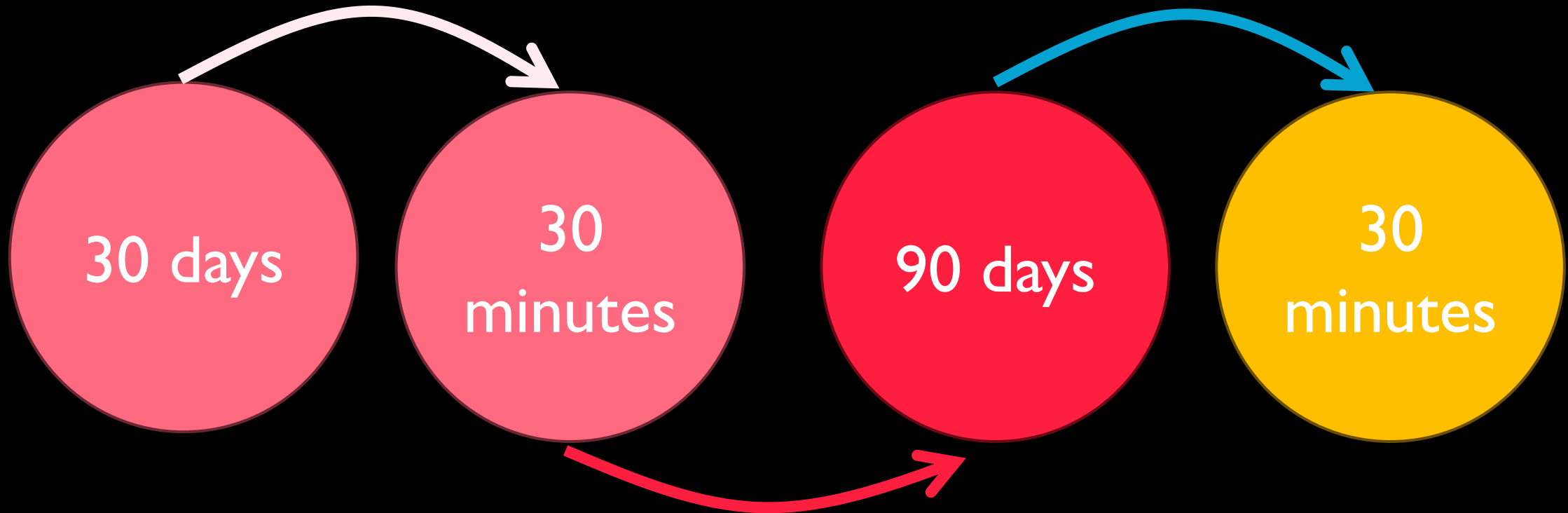https://www.linkedin.com/in/maaret/    @maaretp@mas.to

# With All Test Green in CI Principle, Release Tagging Preserves Tests for Compliance

When the color is green, no one cares about how many pass and fail.

https://www.linkedin.com/in/maaret/     @maaretp@mas.to

# Context Changes Over Time



30 days

30 minutes

90 days

30 minutes

# "Releases are overhead. So much work."

# What is the Work?

Build and release pipeline

Evidence for compliance

Testing for all things in the results gap

Approvals and permissions

What You Have

What You Target

https://www.linkedin.com/in/maaret/    @maaretp@mas.to

# Release Review

**3579**
compliance checks

| Action | |
| --- | --- |
| Architecture and Designs (D1, D6) | Documentation |
| Technical Quality (Q2, Q3) | Linters<br>Unit test coverage |
| System Test Contents (Q1, Q4, Q5, Q6, Q7, D1) | Master test plan, test cases, test summary<br>Sufficient testing |
| Security (Q9) | Threat analysis<br>Vulnerability management |
| IPR and Licenses (Q8, E1, E2) | OS, Middleware, Application inventory<br>Forbidden licenses check |
| Product Perspectives (Q10, Q11, Q12, D3, D4) | Usability<br>Documentation<br>Localization |
| Stakeholder involvement (Q13) | |
| Archiving (D2, D5, D7) | Release repeatability<br>Plans for next up<br>Installation media |

https://www.linkedin.com/in/maaret/   @maaretp@mas.to
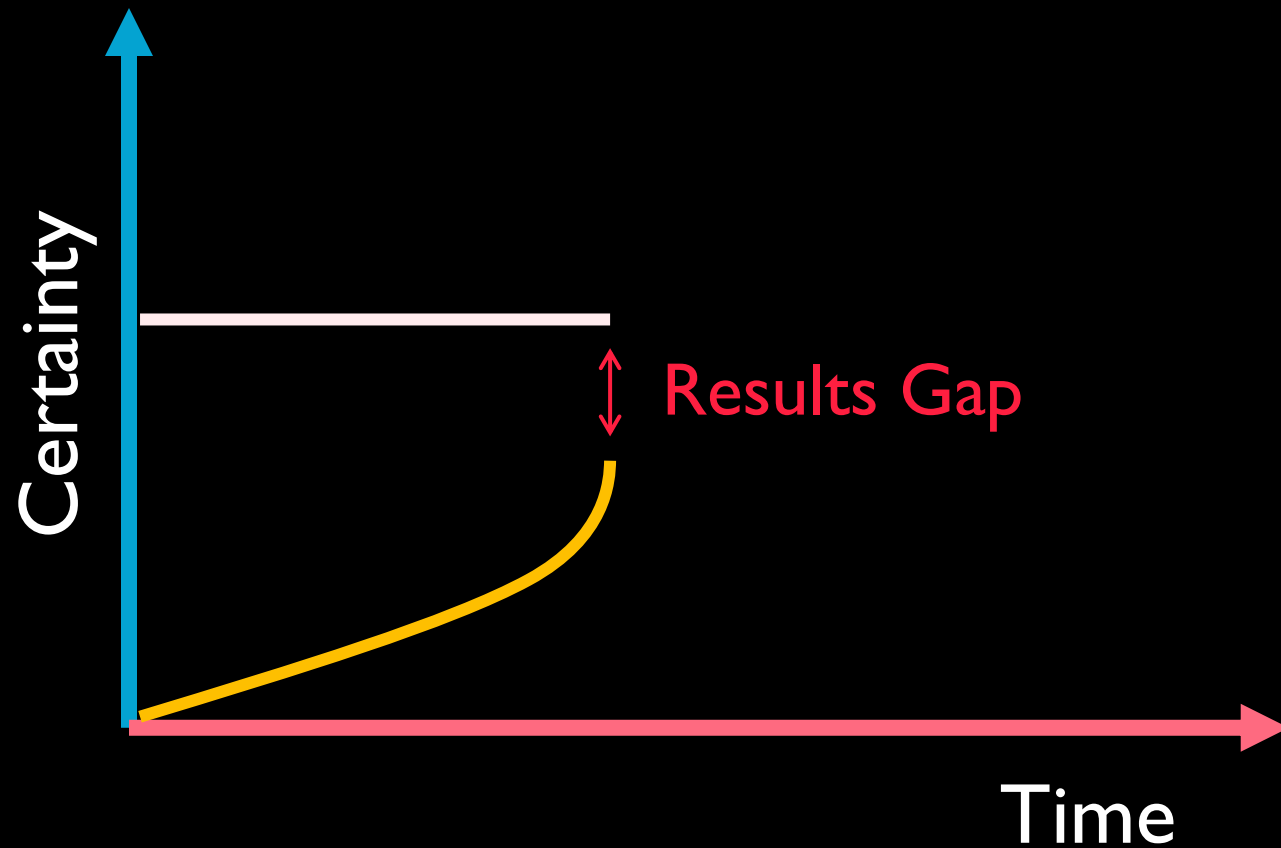
What must be different
to shorten release testing
from 30 days to 30
minutes?

# Fact Checking

There is no automation that gets a month of work done in 30 minutes. Just no.

# Fact Checking

We do less in release testing.
We do more before release
testing.

Fact Checking

We don't have bugs. Fixing delays and causes retesting.
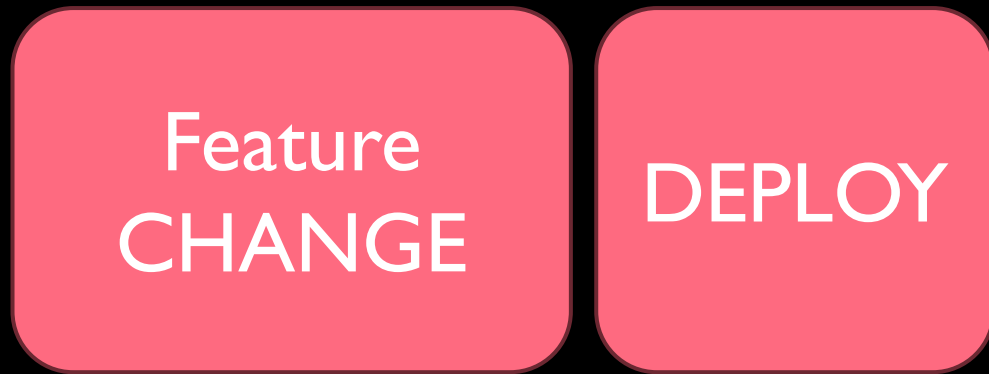We don't wait for testing. Testing can continue after.

https://www.linkedin.com/in/maaret/ @maaretp@mas.to

# Features, Deployments, Releases

Feature flag

**Feature CHANGE** **DEPLOY**

**DEPLOY** **RELEASE**

System testing =   Feature testing +

Release testing

# Guidelines

**EaC**. Nothing changes without changing something. **Follow the changes**.

\* EaC – Everything as Code, particularly Infra

# Guidelines

**Continuous testing.** Test automation alerts on changes. Change calls to explore.

Guidelines

**Pipelines**. Doing the work all the time, with every change. Routine comes with repetition.

Making releases routine is the heartbeat of a good team, creating a bubble of productive serenity.

—Maaret Pyhäjärvi