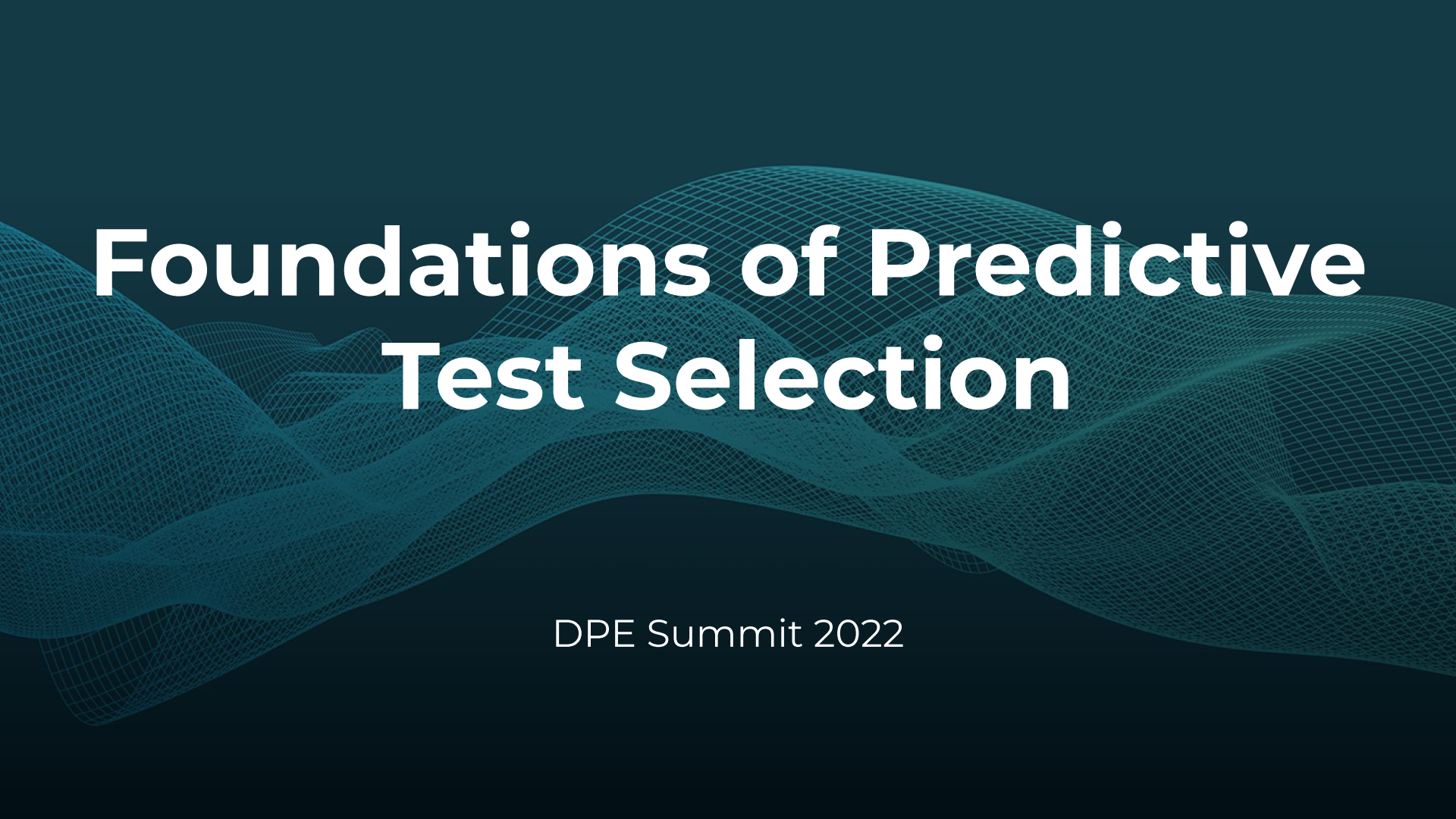




Foundations of Predictive Test Selection

DPE Summit 2022



Eric Wendelin

Principal Data Scientist



Luke Daley

Principal Executive

Test time growth multipliers

1

Developers

2

Tests

3

Automated
changes

4

Functional
complexity

Magnitude of waste

- Most validation time attributable to testing
- Typically less than 1% of codebase affected by any given change
- Typically fewer than 1% of tests affected by any given change
- Yet, most reported test failures are not regressions

Common test time reduction strategies

1

Build Caching

2

CI Fanout

3

Test Distribution

4

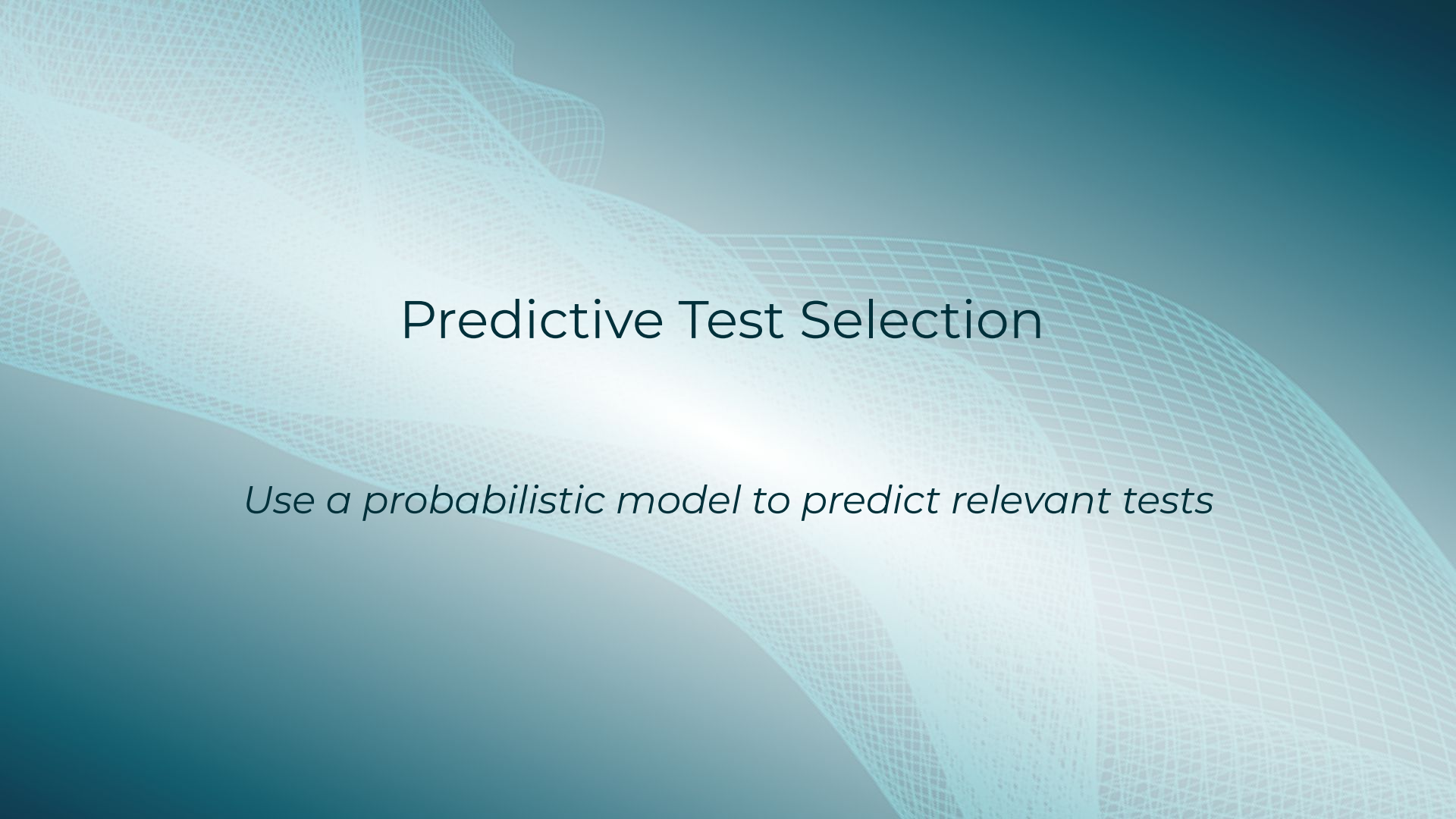
Static/Dynamic Analysis

5

Optimize Slowest

6

Shifting right



Predictive Test Selection

Use a probabilistic model to predict relevant tests

Not a new concept

- “Predictive Test Selection” — Meta 2019
- “Improving Test Effectiveness Using Test Executions History: An Industrial Experience Report” — Ericsson 2019
- “Taming Google-Scale Continuous Testing” — Google 2017
- “Test Re-prioritization in Continuous Testing Environments” — Concordia Univ. 2016
- “The Art of Testing Less without Sacrificing Quality” — Microsoft 2015
- “Improving the effectiveness of test suite through mining historical data” — ACM 2014
- ... and more

Emerging commercial solutions

- Appsurify TestBrain
- Codecov Smart automated test selection
- CQSE Pareto Testing
- Launchable Predictive Test Selection
- Sealights Test Impact Analysis
- ... more in development

Tenets of Predictive Test Selection UX

1

Transparency

2

Control

3

Observability

4

Tool Integration

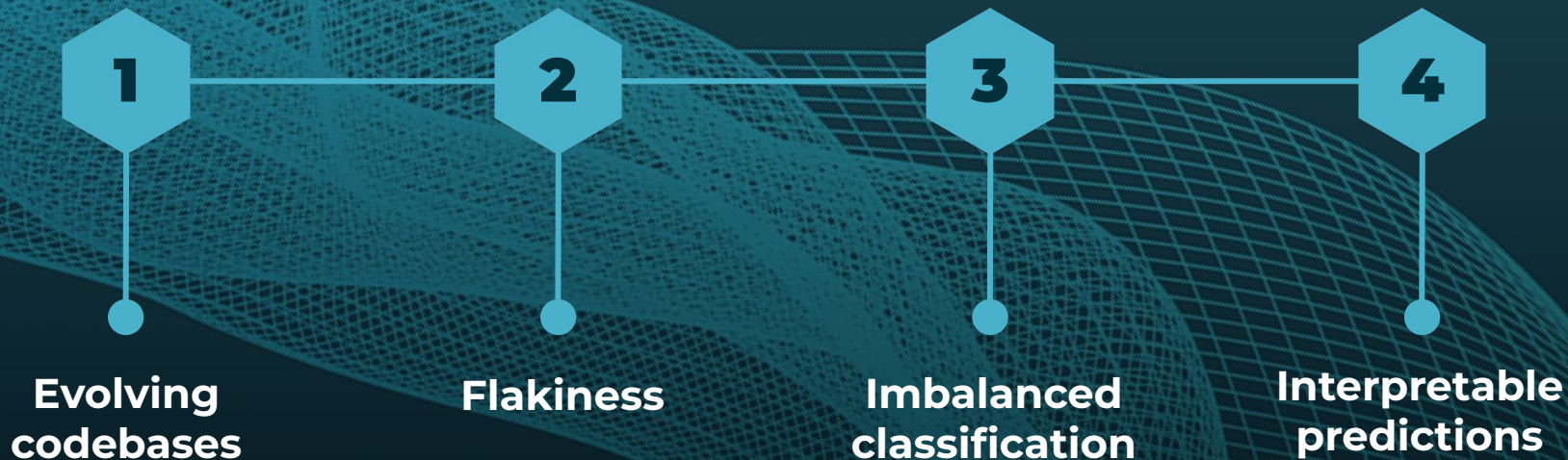
5

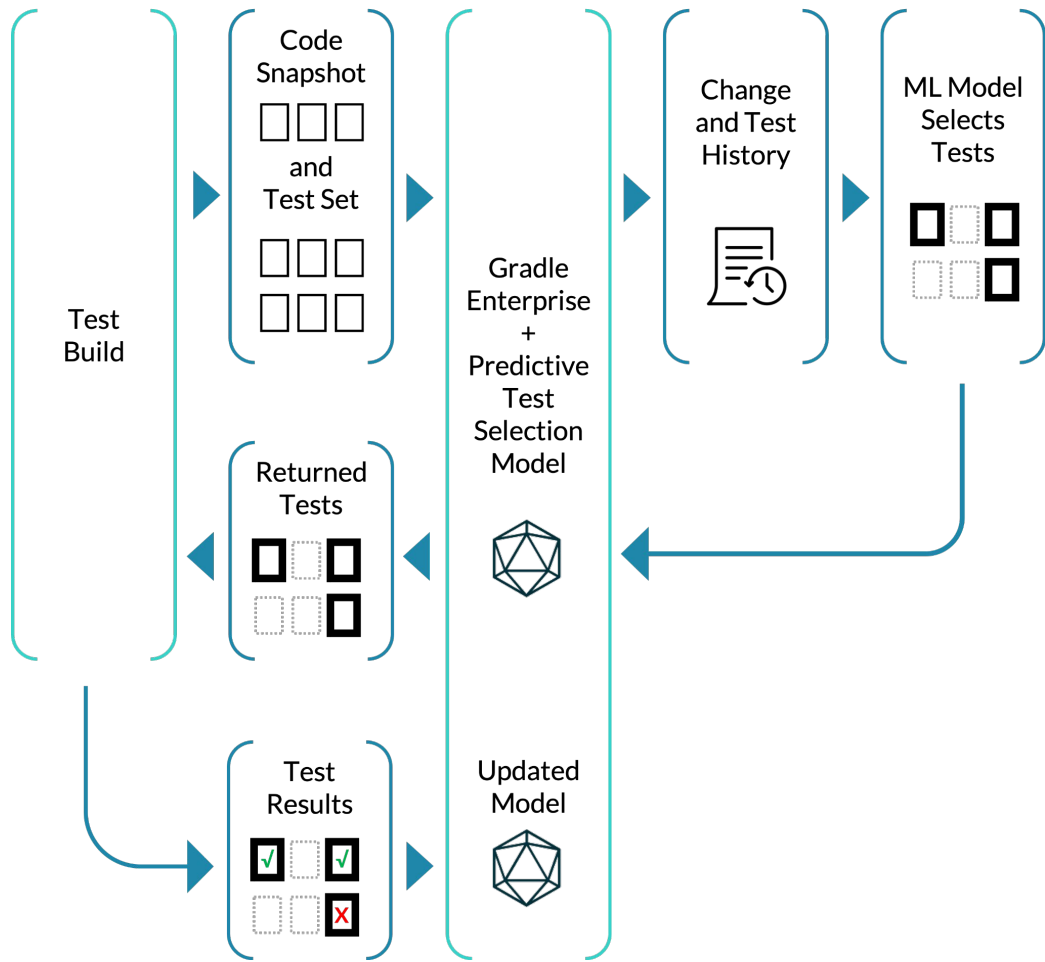
Effectiveness

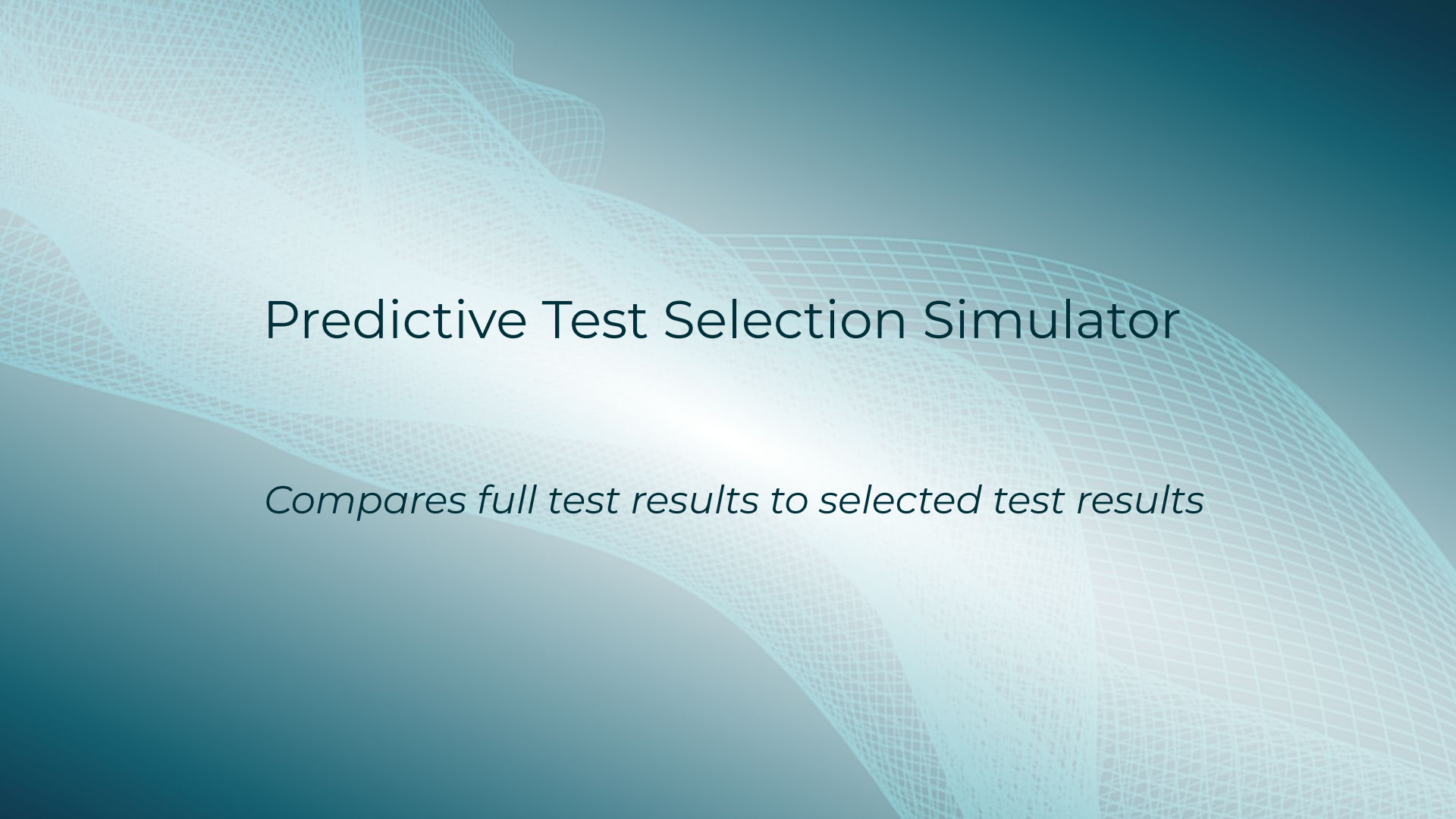
6

Adaptation

Challenges of Predictive Test Selection







Predictive Test Selection Simulator

Compares full test results to selected test results

Predictive Test Selection Simulator

Find test task/goal

Usage (Not enabled)

Simulations (2.40K builds)

Mean build time: N/A

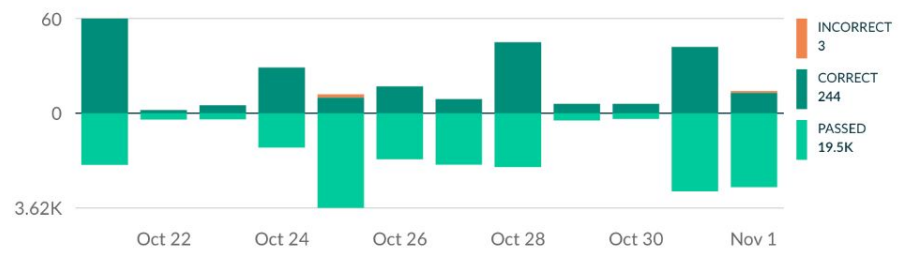
Mean build time: 5 min 54 sec

Task/Goal failures predicted ?

98.8% (244 of 247 total)

Test failures predicted ?

98.8% (329 of 333 total)

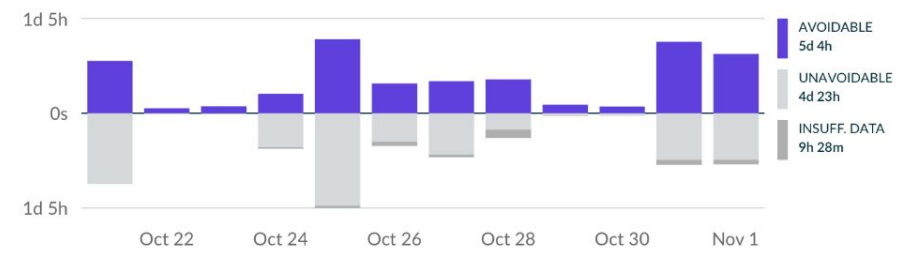


Savings potential ?

5 d 4 hr (49%)

Avoidable tests ?

103K (25%)



Top tasks/goals by mean duration ?

Path	Task/Goal failures predicted ?	Test failures predicted ?	Simulations of total builds ?	Mean duration ?	Total test time ?	Savings potential ?
spring-boot-build > :spring-boot-project:spring-boot-to...	0 / 0	0 / 0	48 / 48 (100%)	26 min 45 sec	21 hr 21 min	14 hr 53 min (70%)
spring-boot-build > :spring-boot-project:spring-boot-aut...	18 / 18 (100%)	18 / 18 (100%)	145 / 145 (100%)	19 min 4 sec	1 d 19 hr	1 d 2 hr (60%)

Predictive Test Selection Simulator

spring-boot-build

:spring-boot-project:spring-boot-test-autoconfigure:test

Usage (Not enabled)

Mean build time: N/A

Simulations (146 builds)

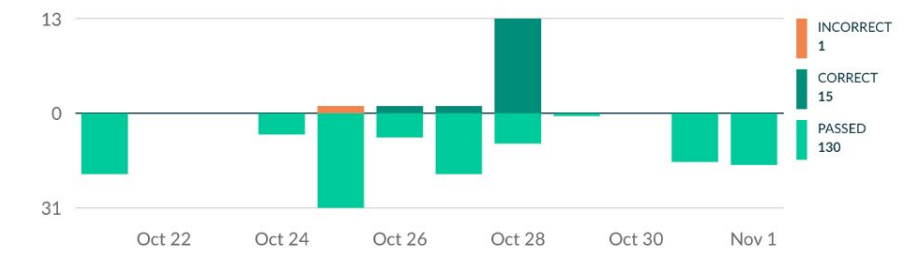
Mean build time: 27 min 15 sec

Task/Goal failures predicted

93.8% (15 of 16 total)

Test failures predicted

96.3% (26 of 27 total)

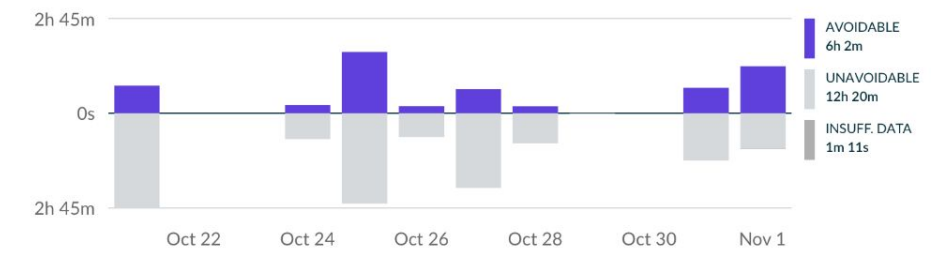


Savings potential

6 hr 2 min (33%)

Avoidable tests

3.96K (24%)



Possibly incorrect predictions

Build start time	Outcome	Test failures predicted	Avoidable tests	Savings potential	Hostname
Oct 25 2022 at 11:18:36 AM	FAILED	0 / 1	6 / 19	50m 24s / 51m 22s	sfrederick
Local JDK-17 Linux gh-32829 dirty					
1 possibly incorrect prediction: org.springframework.boot.test.autoconfigure.data.elasticsearch.DataElasticsearchTestIntegrationTests					

- Summary
- Console log
- Failure
- Deprecations
- Timeline
- Performance

Tests

- Projects
- Dependencies
- Build dependencies
- Plugins
- Custom values
- Switches
- Infrastructure

- See before and after
- Compare Build Scan

Overview

Slowest tests

:spring-boot-project:spring-boot-test-autoconfigure:test

Test Class (use * to match zero or more characters)

FAILED

Test (use * to match zero or more characters)

Found 1 test executed in 1 test class taking 0.033s, 1 failed

Test	Outcome ?	Execution time ?
:spring-boot-project:spring-boot-test-au...	FAILED	3.970s
org.springframework.boot.test.autoconfi..	FAILED	0.033s
initializationError	FAILED	0.033s



ge.spring.io



Summary



Console log



Failure



Deprecations



Timeline



Performance



Tests



Projects



Dependencies



Build dependencies



Plugins



Custom values



Switches



Infrastructure



See before and after



Compare Build Scan

initializationError FAILED`:spring-boot-project:spring-boot-test-autoconfigure:test > org.springframework.boot.test.autoconfigure.data.elasticsearch.DataElasticsearchTest...`0 results in last 7 days, 0 passed [View test history](#)

11726 tests in 1530 test classes in build, 2 failed

Execution 1 of 1 FAILED 0.033s Started Oct 25 2022 at 12:12:21 PM PDT

```
▶ at org.testcontainers.containers.GenericContainer.doStart(GenericContainer.java:349)
...

```

Caused by: org.rnorth.ducttape.RetryCountExceededException: Retry limit hit with exception

```
▶ at org.rnorth.ducttape.unreliables.Unreliables.retryUntilSuccess(Unreliables.java:88)
...

```

Caused by: org.testcontainers.containers.ContainerLaunchException: Could not create/start container

```
▶ at org.testcontainers.containers.GenericContainer.tryStart(GenericContainer.java:542)
...

```

Caused by: java.lang.IllegalStateException: Container crashed with out-of-memory (OOMKilled)

```
▶ at org.testcontainers.containers.GenericContainer.tryStart(GenericContainer.java:505)
...

```

Output*This execution did not produce any output.*

Predictive Test Selection Usage Dashboard

Measures adoption and impact



Predictive Test Selection › micronaut › :http-server-netty:test

Usage (393 builds)

Simulations (184 builds)

Mean build time: 12 min 54 sec

Mean build time: 28 min

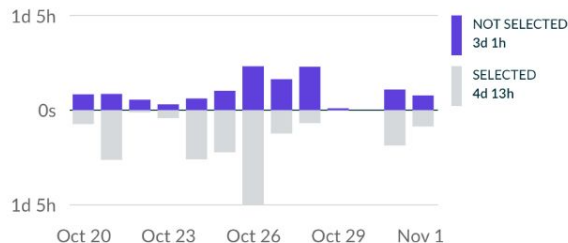
Test tasks/goals which enabled Predictive Test Selection

393 (68% of total)



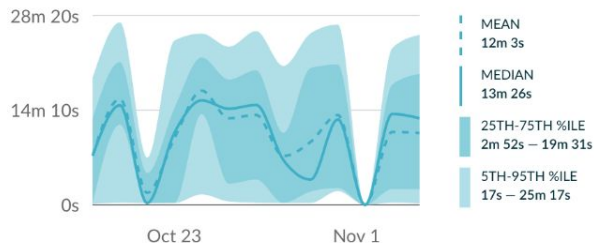
Serial test time saved

3 d 1 hr (62% of 4 d 23 hr total savings potential)



Mean duration

12 min 3 sec (65% faster with test selection enabled)



Recent builds

Build start time	Outcome	Duration	Not selected test classes	Serial test time saved	Hostname
yesterday at 11:56:00 PM	PASSED	16 min 29 sec	9/123	4 min 36 sec/41 min 56 sec	fv-az573-153
Linux CI Predictive Test Selection HEAD github:action					
yesterday at 11:53:49 PM	PASSED	13 min 45 sec	9/123	4 min 36 sec/36 min 34 sec	fv-az246-965
Linux CI Predictive Test Selection HEAD github:action					

-  Summary
 -  Console log
 -  Failure
 -  Deprecations
 -  Timeline
 -  Performance
 -  **Tests**
 -  Projects
 -  Dependencies
 -  Build dependencies
 -  Plugins
 -  Custom values
 -  Switches
 -  Infrastructure
-
-  See before and after
 -  Compare Build Scan

Overview

Slowest tests



   Found 184 tests executed in 15 test classes taking 2m 33.698s

Test	Outcome 	Execution time 
:http-server-netty:test >	 PASSED	2m 33.698s
io.micronaut.http.server.netty.types.FileTypeHandlerSpec		
io.micronaut.http.server.netty.util.DefaultHttpHostSpec >		
io.micronaut.http.server.netty.binding.JsonBodyBinding...		
io.micronaut.http.server.netty.stack.InvocationStackSpec		
io.micronaut.http.server.netty.binding.QueryParameterF...		
io.micronaut.http.server.netty.errors.CustomErrorUnsati...		
io.micronaut.http.server.netty.binding.ParameterBinding...		
io.micronaut.http.server.netty.cors.CorsFilterSpec >	 PASSED	10.339s
io.micronaut.http.server.netty.java.ParameterBindingSpec	 PASSED	9.592s
io.micronaut.http.server.netty.RequestLineSpec >	 PASSED	9.420s
io.micronaut.http.server.netty.reactivesequence.Reactiv...	 PASSED	3.524s
io.micronaut.docs.context.annotation.primary.PrimaryS...	 SKIPPED	0.001s
io.micronaut.docs.http.server.exception.ExceptionHandl...	 NOT SELECTED	0.000s
io.micronaut.docs.http.server.exception.SimpleControlle...	 NOT SELECTED	0.000s
io.micronaut.docs.http.server.exception.aterrorexceptio...	 NOT SELECTED	0.000s
io.micronaut.docs.respondingnotfound.BooksSpec	 NOT SELECTED	0.000s



 Predictive Test Selection result

High probability test affected by recent changes

[View test history](#)

Turning it on

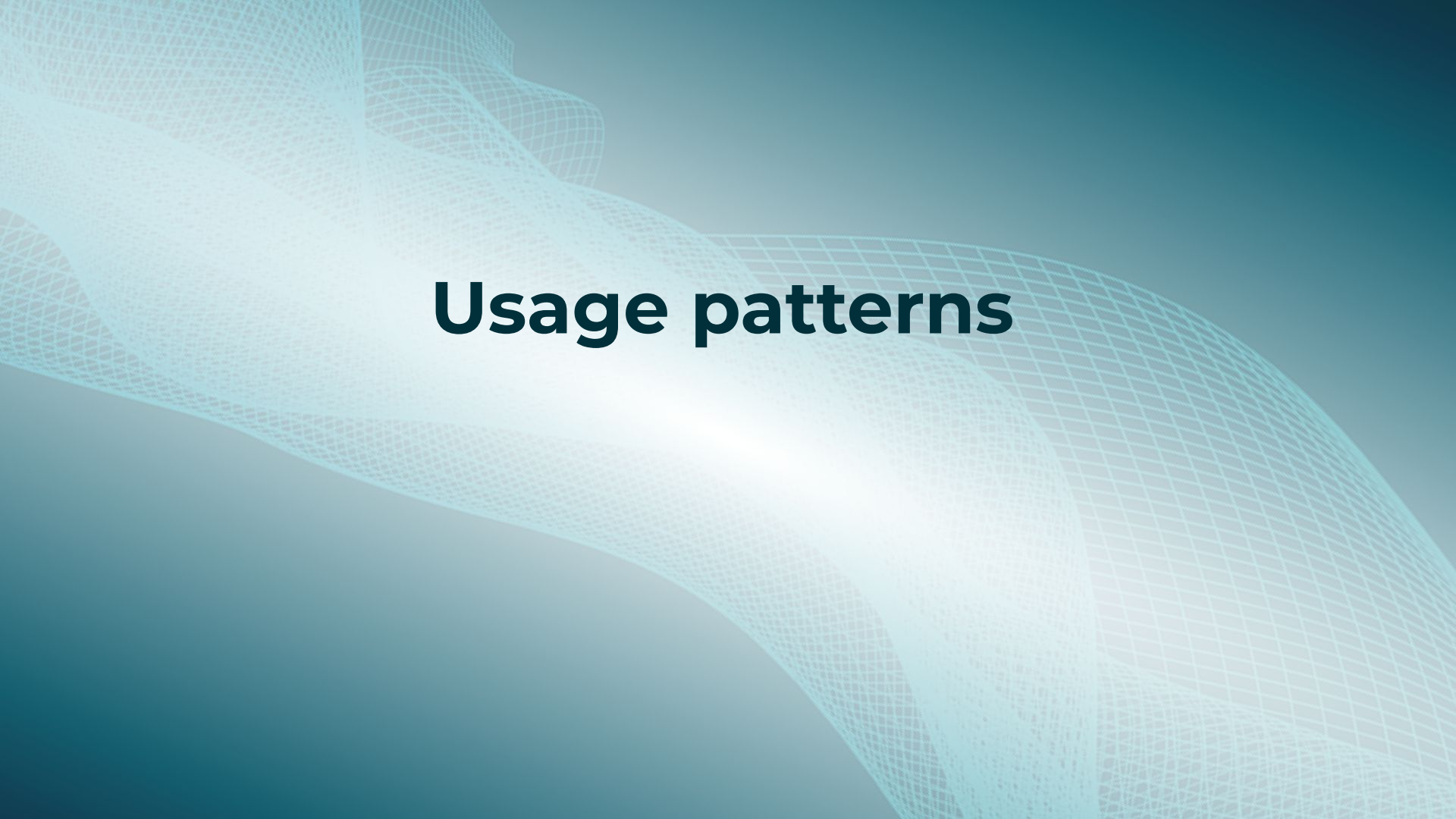
```
// settings
plugins {
    id("com.gradle.enterprise") version "3.10.3"
}

gradleEnterprise {
    server = "https://gradle-enterprise.mycompany.com"
}

// project
tasks.test {
    predictiveSelection {
        enabled.set(true)
    }
}
```

Must-run tests

```
tasks.test {  
    predictiveSelection {  
        enabled.set(true)  
        mustRun {  
            includeAnnotationClasses.add("com.project.annotations.MustRun")  
        }  
    }  
}
```

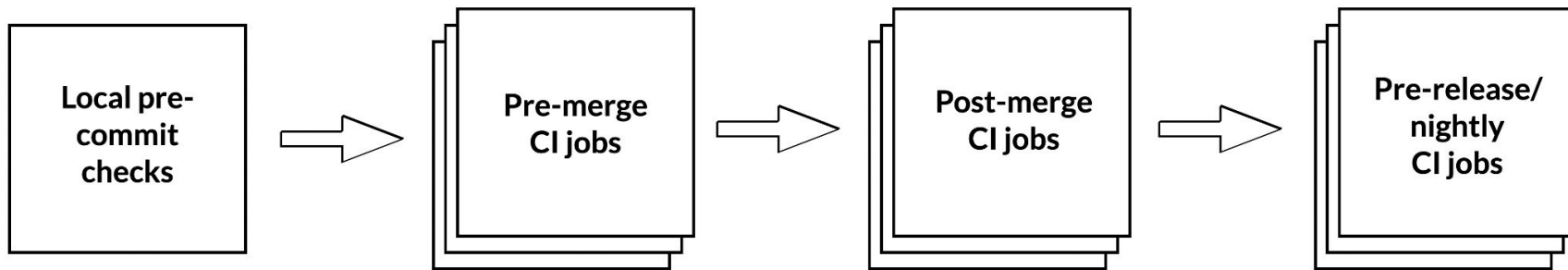
The background features a series of overlapping, wavy, translucent bands in shades of teal and white. These bands are overlaid with a fine, white grid pattern, creating a sense of depth and movement. The overall color palette is a gradient of teal, ranging from a darker blue-green at the top to a lighter, almost white teal at the bottom.

Usage patterns

Shift right to shift left

Apply to existing pre-merge verification

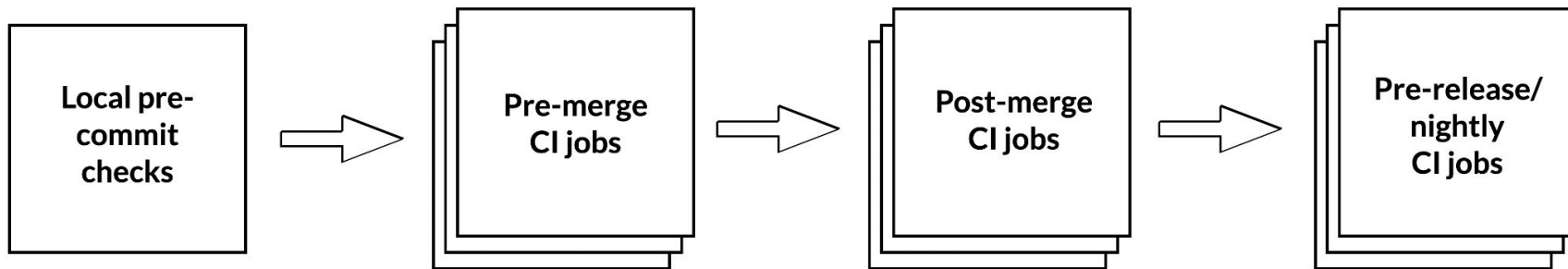
1. Most common pattern
2. Faster *overall* feedback - reduce CI pressure
3. Easy to adopt - can usually just turn it on
4. Have a plan for dealing with post-merge failures



Shift *further* right to shift left

Apply to existing pre and post merge verification

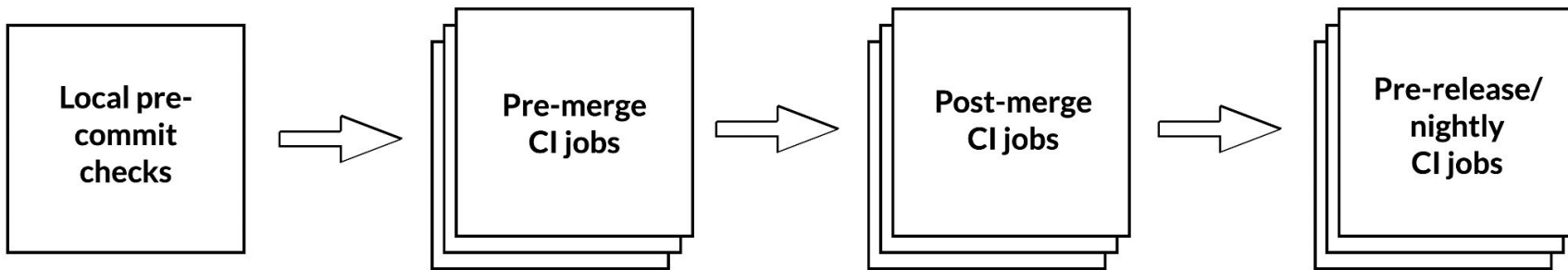
1. Move “all the tests” to a nightly build
2. Offers the most overall time saving
3. Delays feedback time for worst case



Shift left

Run some high value tests pre-merge

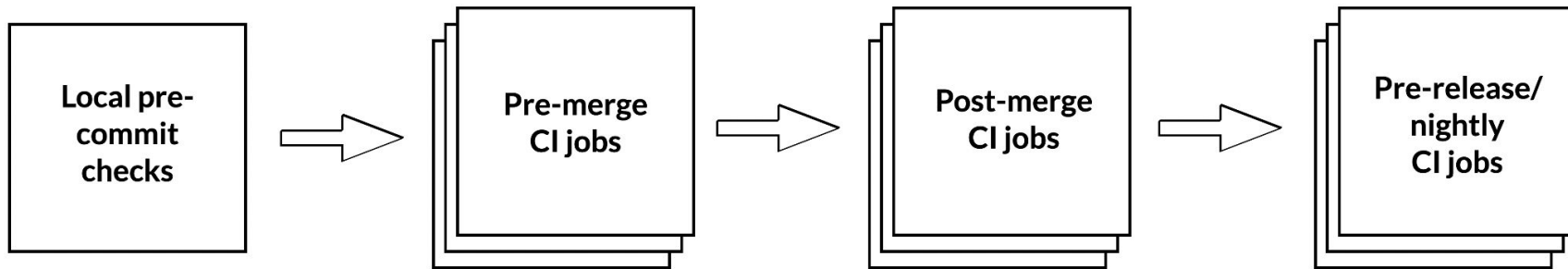
1. Applicable when integration/functional/slow tests are only run late
2. Faster individual change feedback
3. Requires changing build plans
4. May require more CI resources
5. Automatic canary testing



Shift slightly left

Two-pass pre-merge

1. Split pre-merge verification into two steps
2. Faster feedback through fail-fast
3. Requires changing build plans





Local builds?

Is Predictive Test Selection for you?

1. Consider Build Caching
2. Consider Test Distribution
3. If shifting right: can your team action missed failures?
4. How flaky are your tests?

Adopting Predictive Test Selection

1. Use the simulator to gauge effectiveness for your project
2. Identify projects and tasks with a lot to gain
3. Verify that missed prediction rate is acceptable
4. Establish your plan for handling missed failures
5. Use the dashboard to monitor effectiveness once applied

THANK YOU

Curious about the future?

Discuss with Eric, Luke, and others in
the Gradle Lounge

Icons



Predictive Test Selection

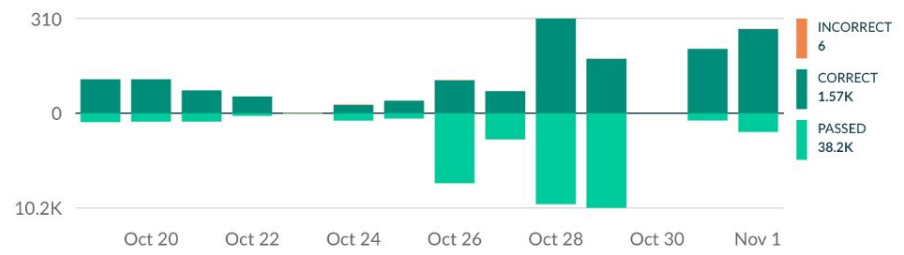
Find test task/goal

Usage (3.06K builds)
Mean build time: 8 min 34 sec

Simulations (30.3K builds)
Mean build time: 2 min 8 sec

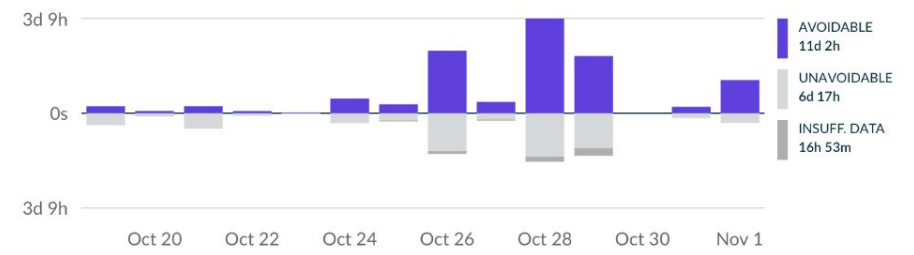
Task/Goal failures predicted 99.6% (1.57K of 1.58K total)

Test failures predicted 99.7% (6.08K of 6.10K total)



Savings potential 11 d 2 hr (60%)

Avoidable tests 145K (43%)



Top tasks/goals by mean duration

Path	Task/Goal failures predicted	Test failures predicted	Simulations of total builds	Mean duration	Total test time	Savings potential
micronaut-starter > :test-features:test	0 / 0	0 / 0	42 / 97 (43.3%)	1 hr 53 min	2 d 17 hr	1 d 7 hr (48%)
micronaut-starter > :test-aws:test	0 / 0	0 / 0	41 / 71 (57.7%)	1 hr 15 min	1 d 18 hr	1 d 5 hr (69%)