



Say it works on my machine
one more time...
I dare you!

Oleg Šelajev

DEVOXX™
POLAND

What actually drives
productivity?

What actually drives productivity?

The developer-centric approach to measuring and improving productivity.

Abi Noda, DX

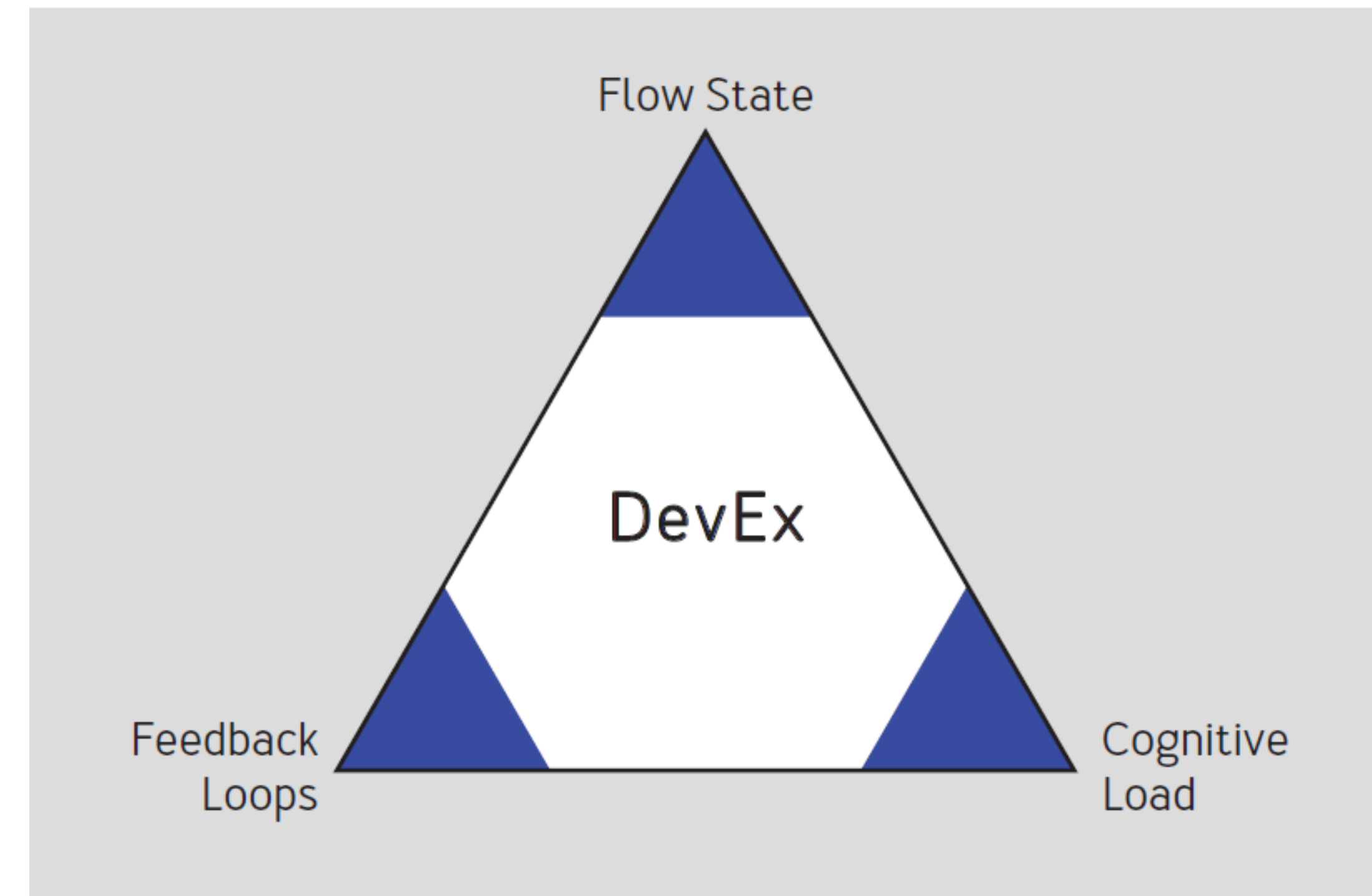
Margaret-Anne Storey, University of Victoria

Nicole Forsgren, Microsoft Research

Michaela Greiler, DX

<https://queue.acm.org/detail.cfm?id=3595878>

FIGURE 1: **THREE CORE DIMENSIONS OF DEVELOPER EXPERIENCE**



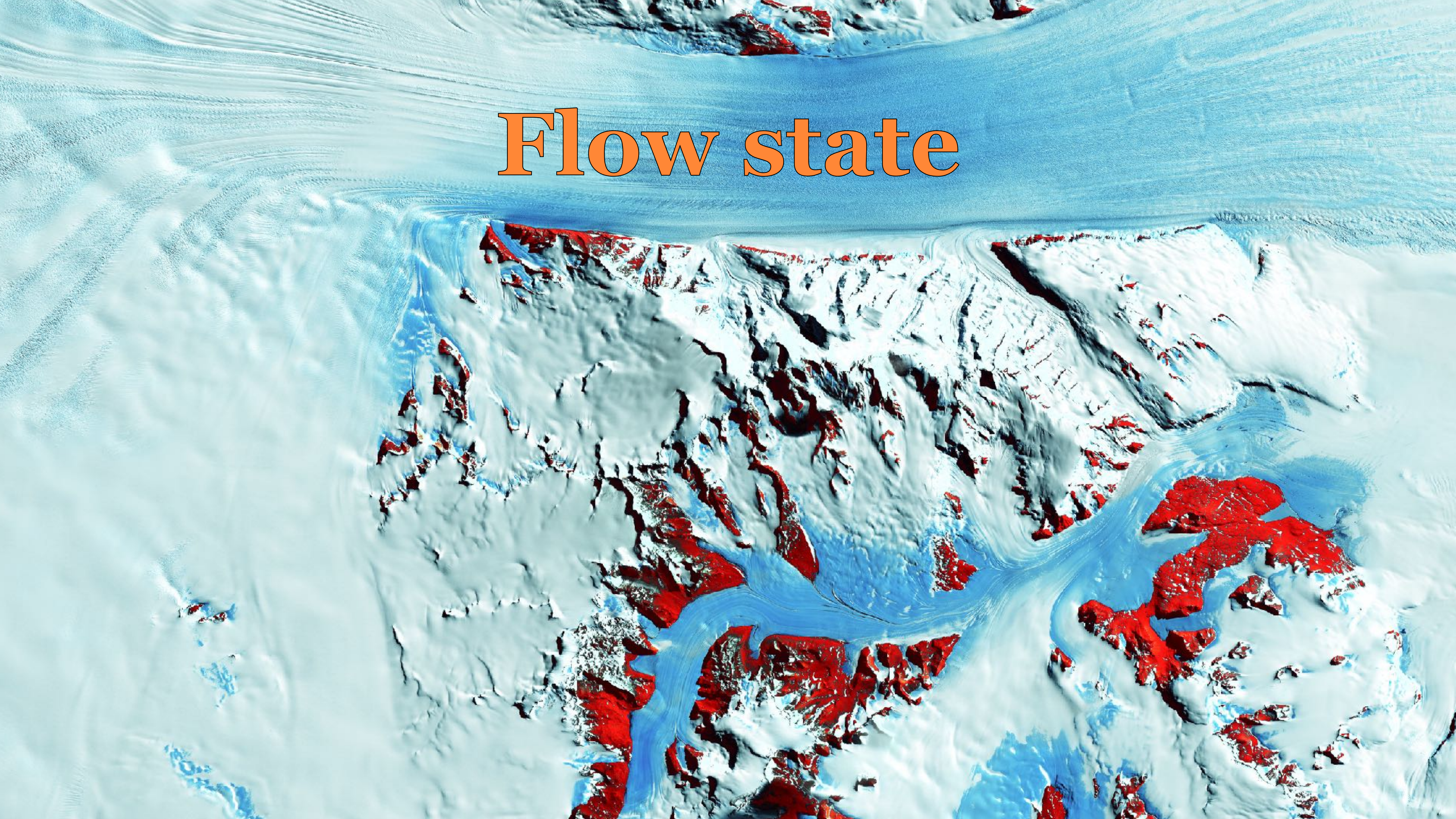
Feedback flows



Cognitive load



Flow state





Oleg Šelajev
Developer relations



AtomicJar: leader in developer first testing

The company behind developers' most popular multi-language testing framework that is growing at 6M+ Docker Pulls a month and can be found in almost every tech-enabled company.

Testcontainers is used by



Uber



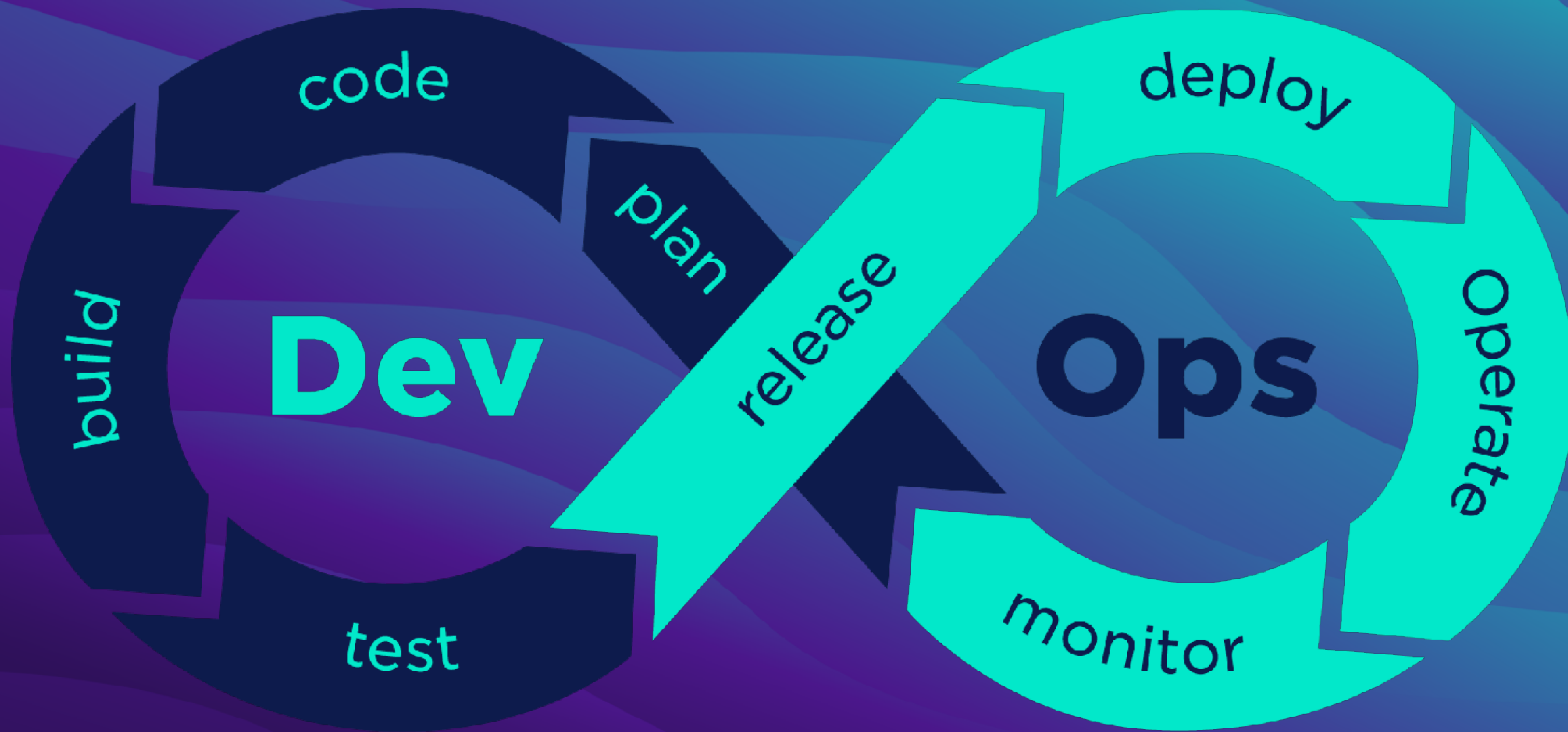
NETFLIX



INSTANA
an IBM Company



Tests are essential



Great tests improve

Feedback loops

Flow state

Cognitive load



A glowing lightbulb hanging from a string of lights against a dark blue background with bokeh effects. The lightbulb is the central focus, with its filament glowing yellow. The background is filled with out-of-focus lights in shades of blue and yellow, creating a bokeh effect. The text is overlaid on the image in a bold, white, sans-serif font.

**With bad tests nothing
will help your developer
productivity***



Can't run locally
Break on changes
Flaky runs
Slow pipelines



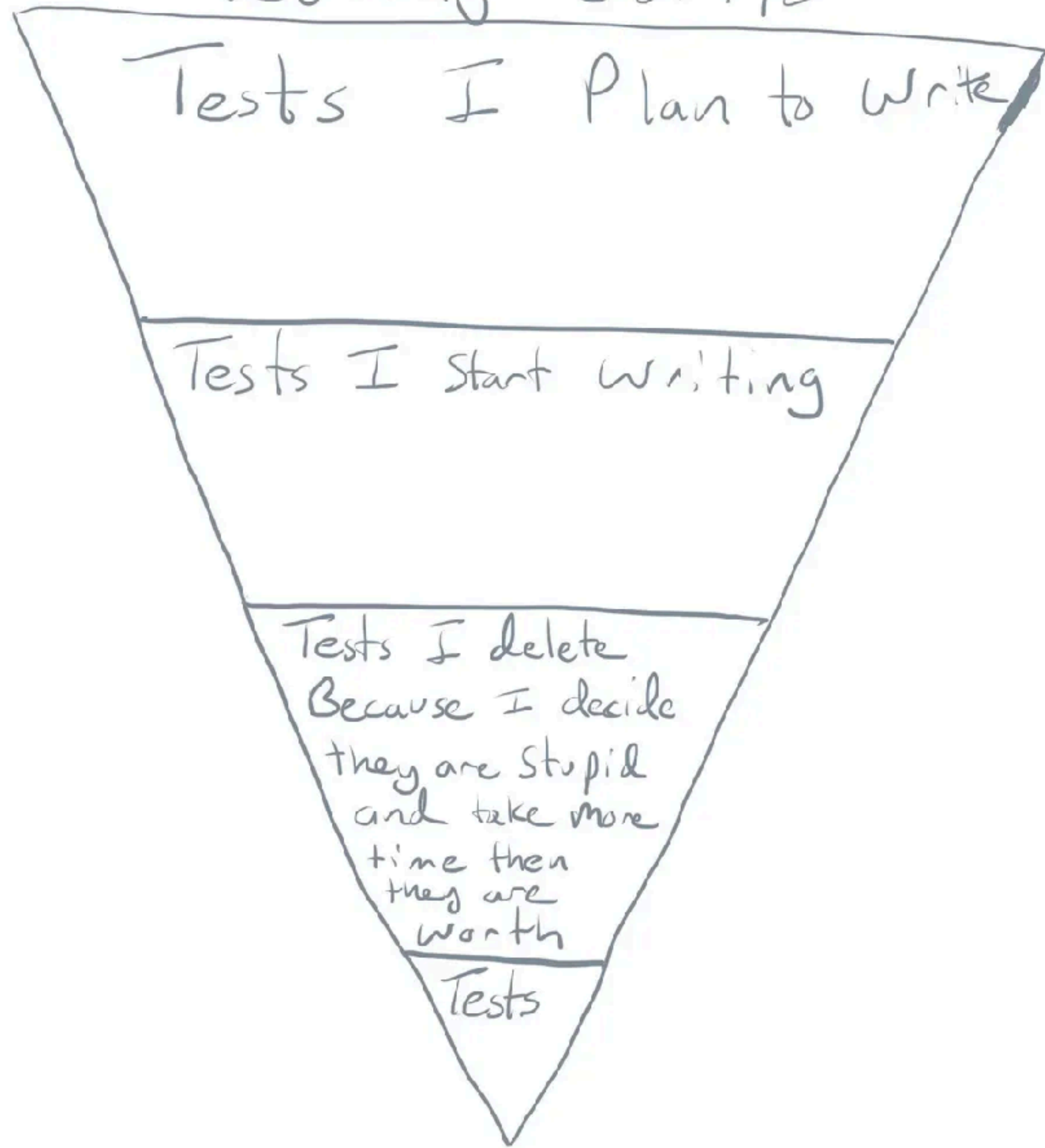


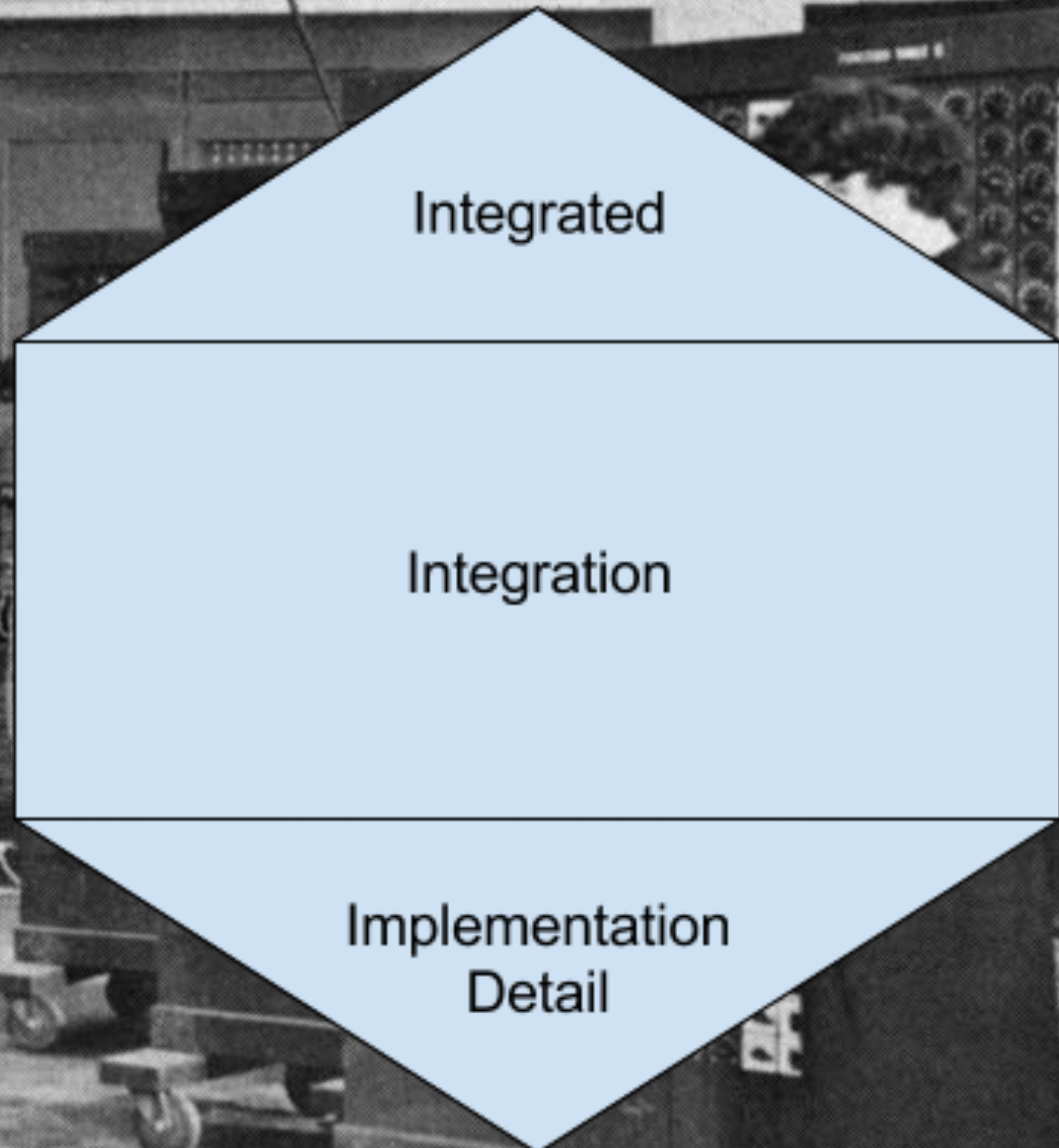
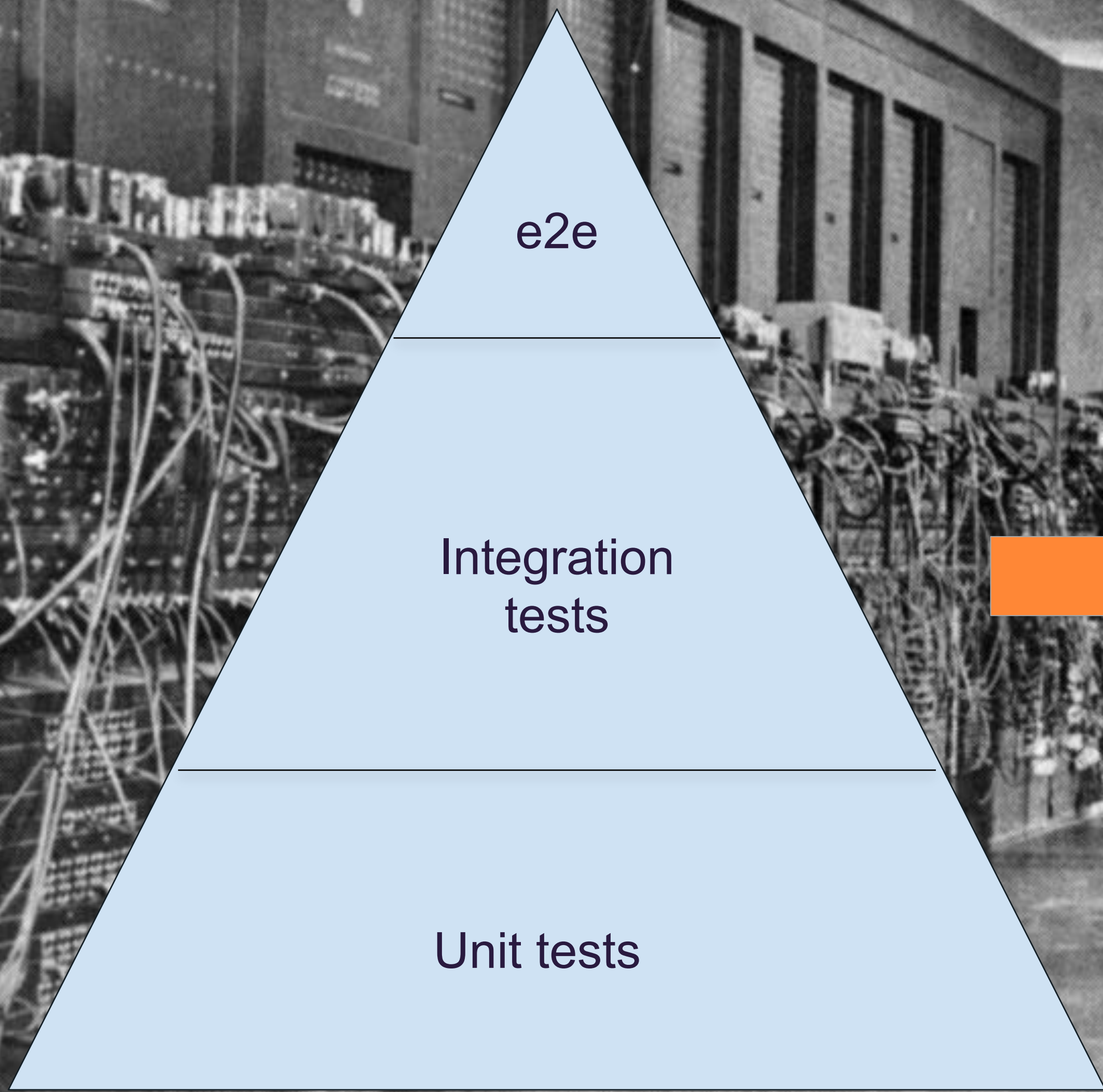
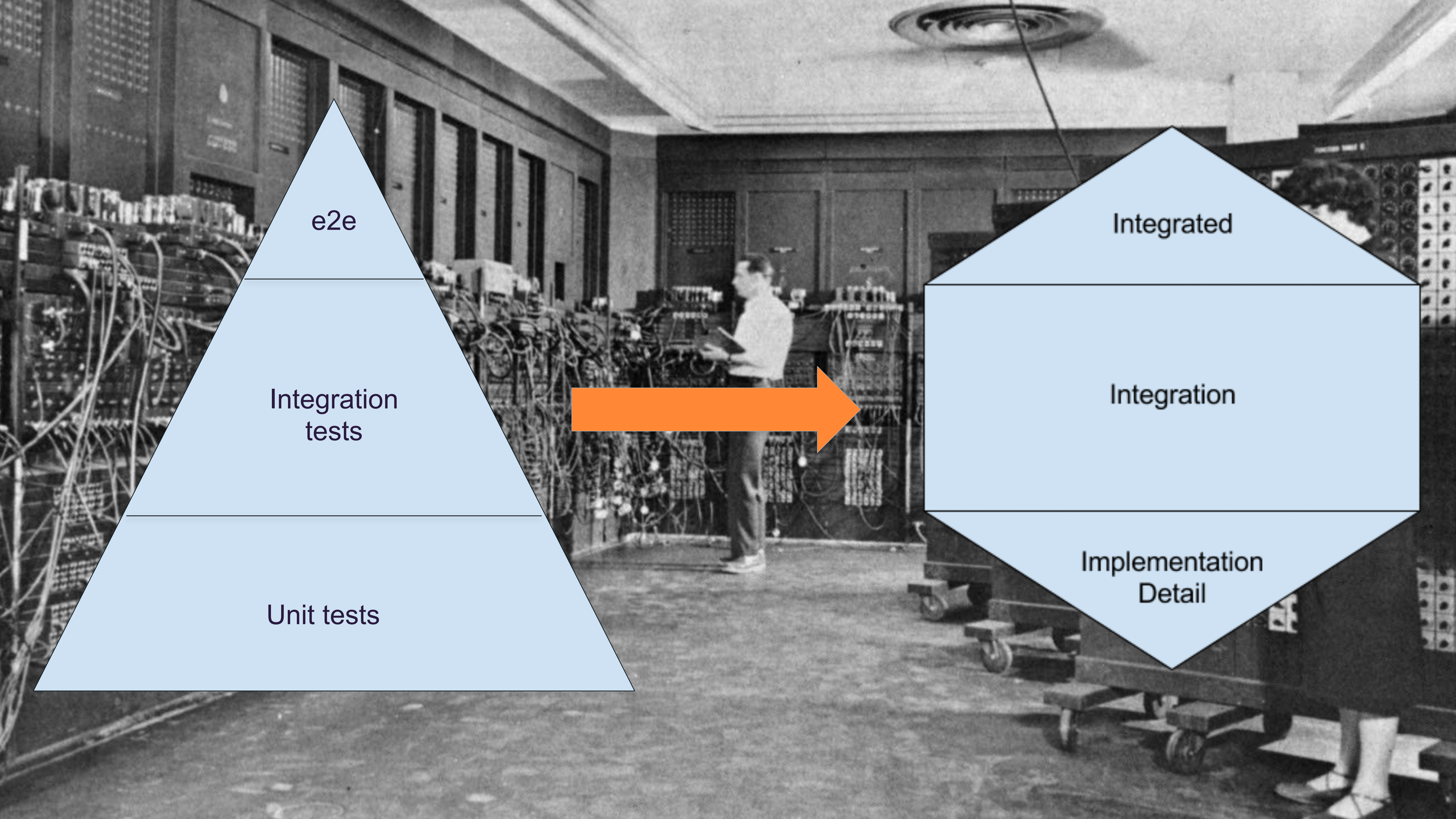
Novel
Reproducible
Significant



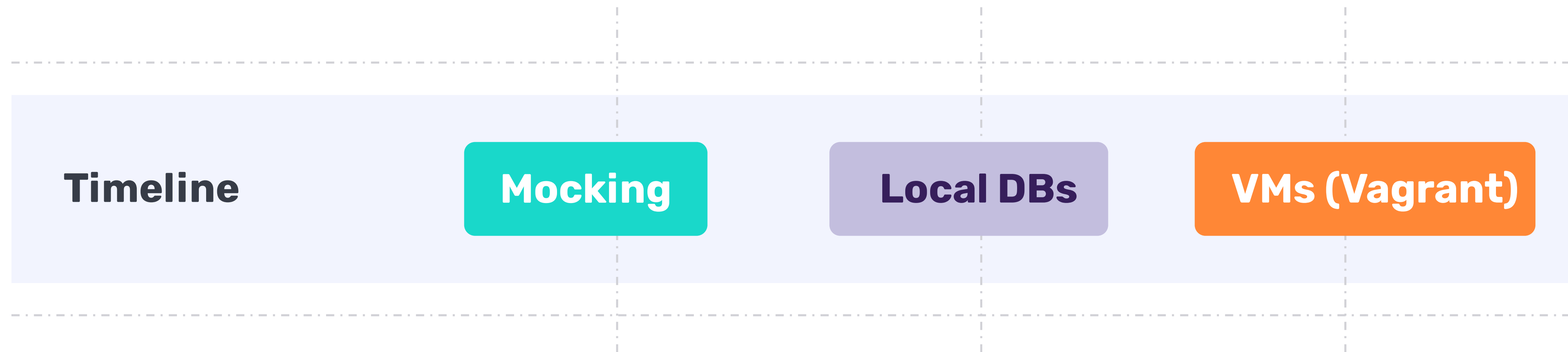
How to do tests?

Testing Dorito

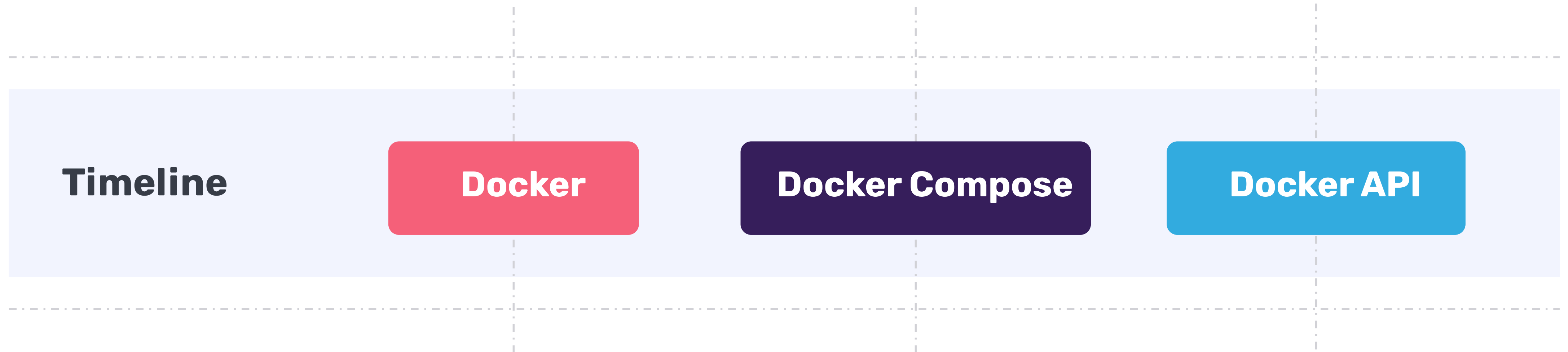




Integration Testing transformation over the years



Integration Testing transformation over the years





mosheeshel / DockerContainerRule.java

Created 7 years ago • Report abuse

<> Code

Revisions 1

Stars 15

Forks 3

JUnit @rule for starting a docker container from within an Integration test

<> DockerContainerRule.java

```
1 package rules;
2
3 import com.google.common.collect.Lists;
4 import com.google.common.collect.Maps;
5 import com.spotify.docker.client.DefaultDockerClient;
6 import com.spotify.docker.client.DockerClient;
7 import com.spotify.docker.client.DockerException;
8 import com.spotify.docker.client.messages.ContainerConfig;
9 import com.spotify.docker.client.messages.ContainerCreation;
10 import com.spotify.docker.client.messages.HostConfig;
11 import com.spotify.docker.client.messages.PortBinding;
12 import org.eclipse.jdt.launching.SocketUtil;
13 import org.junit.rules.ExternalResource;
14 import org.junit.runner.Description;
15 import org.junit.runners.model.Statement;
16
```

- Container lifecycle & cleanup
- Container & service configuration
- Integration with frameworks or tests



mosheeshel / DockerContainerRule.java

Created 7 years ago • Report abuse

<> Code

Revisions 1

Stars 15

Forks 3

JUnit @rule for starting a docker container from within an Integration test

<> DockerContainerRule.java

```
1 package rules;
2
3 import com.google.common.collect.Lists;
4 import com.google.common.collect.Maps;
5 import com.spotify.docker.client.DefaultDockerClient;
6 import com.spotify.docker.client.DockerClient;
7 import com.spotify.docker.client.DockerException;
8 import com.spotify.docker.client.messages.ContainerConfig;
9 import com.spotify.docker.client.messages.ContainerCreation;
10 import com.spotify.docker.client.messages.HostConfig;
11 import com.spotify.docker.client.messages.PortBinding;
12 import org.eclipse.jdt.launching.SocketUtil;
13 import org.junit.rules.ExternalResource;
14 import org.junit.runner.Description;
15 import org.junit.runners.model.Statement;
16
```

- Container lifecycle & cleanup

- Container & service configuration



huksley commented on Nov 8, 2017

Superseded by <https://www.testcontainers.org/>



Testcontainers

Unit tests with real dependencies



TEST DEPENDENCIES AS CODE

Get lightweight and throwaway containers during your tests



FOR YOUR ENTIRE STACK

Test against any container image: database, message broker, browser, etc.



FOR ANY LANGUAGE

Get started with the open source library for any of 6+ languages

WE ❤️ OPEN SOURCE
& EMPLOY MAINTAINERS



Couchbase



docker
Compose



cassandra



ClickHouse



DB2



Solr



MongoDB



CockroachDB



Microsoft
SQL Server



MySQL



influxdb



Selenium



MariaDB



QuestDB



neo4j



PostgreSQL



OrientDB



HashiCorp
Vault



TiDB



presto



trino



Microsoft
Azure



elasticsearch



Redpanda



HashiCorp
Consul



Google Cloud



HIVEMQ



K3S



RabbitMQ



PULSAR



kafka



LocalStack



NGINX
Part of F5

Modules

Testcontainers modules are preconfigured implementations of various dependencies that make writing your tests even easier!














 [Clear Filter](#)  Official Modules

Languages

- All
- .NET
- Go
- Java
- Node.js

Categories

- All

 ArangoDB NoSQL Database Java Node.js	 Azure Cosmos DB Cloud Java	 Azure SQL Edge Cloud .NET
 Cassandra NoSQL Database Java	 ClickHouse Relational Database Java	 CockroachDB  Relational Database Java
 Consul Other Java	 Couchbase NoSQL Database Java Go .NET	 CouchDB NoSQL Database .NET
 DB2 Relational Database	 Dynamalite NoSQL Database	 DynamoDB NoSQL Database

Modules

Testcontainers modules are preconfigured implementations of various dependencies that make writing your tests even easier!











[Clear Filter](#)  Official Modules

Languages

 All .NET Go Java Node.js

Categories

 All

 CockroachDB Relational Database Java 	 LocalStack Cloud Java Go .NET 	 Neo4j NoSQL Database Java Go .NET Node.js 
 Pulsar Message Broker Java Go 	 Redpanda Message Broker Java .NET 	 YugabyteDB Relational Database Java 

Why are Testcontainers - 🏆🏆🏆

- Flexible lifecycle API: isolation vs convenience
- Capable of abstractions (modules)
- Waiting strategies
- Complex network topologies
- Automatic cleanup
- Everything is dynamic allowing parallelisation



Feels like a unit, reliable like integration

Environment parity



Dev



Prod



Dev



CI



Prod



Testcontainers



GitLab

- Using Docker-in-Docker (DinD)
- Testcontainers Cloud setup
- Parallelisation

The screenshot displays the GitLab CI/CD interface for a project named 'Testcontainers Showcase'. The left sidebar shows navigation options like Project information, Repository, Issues, Merge requests, CI/CD, Pipelines, Editor, Jobs, Schedules, Security & Compliance, Deployments, Packages and registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The main area shows a job log for a job named 'Complete Raw'. The log content includes:

```
Showing last 499.95 KIB of log - Complete Raw
# 11_0.order_id=?
4467 Hibernate: select 11_0.order_id,11_0.id,11_0.product_code,11_0.product_name,11_0.product_price,11_0.quantity from order_items 11_0 where
# 11_0.order_id=?
4468 788A98 [INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.104 s - in org.testcontainers.backstore.orders.domain.Ord
rdarRepositoryTests
4469 782918 [INFO]
4470 782920 [INFO] Results:
4471 782920 [INFO]
4472 782921 [INFO] Tests run: 241, Failures: 0, Errors: 0, Skipped: 0
4473 782921 [INFO]
4474 782954 [INFO]
4475 782956 [INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ testcontainers-showcase ---
4476 783590 [INFO] Building jar: /builds/sivalabs/testcontainers-showcase/target/testcontainers-showcase-0.0.1-SNAPSHOT.jar
4477 784036 [INFO]
4478 784037 [INFO] --- spring-boot-maven-plugin:3.0.0:repackage (repackage) @ testcontainers-showcase ---
4479 785437 [INFO] Replacing main artifact with repackaged archive
4480 785438 [INFO] -----
4481 785438 [INFO] BUILD SUCCESS
4482 785439 [INFO] -----
4483 785445 [INFO] Total time: 13:03 min
4484 785445 [INFO] Finished at: 2023-01-18T14:05:21Z
4485 785446 [INFO] -----
4486 Saving cache for successful job
4487 Creating cache default-non_protected...
4488 .n2/repository: found 3829 matching files and directories
4489 Uploading cache.zip to https://storage.googleapis.com/gitlab-ci-runners-cache/project/41998775/default-non_protected
4490 Created cache
4492 Cleaning up project directory and file based variables
4494 Job succeeded
```


Parallelization

- Surefire plugin

- `parallel=classes`
- `forkCount=N`
- `reuseForks=true`

- Showcase

- `github.com/testcontainers/testcontainers-showcase`

```
[INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0,
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 241, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:47 min
[INFO] Finished at: 2023-04-20T14:26:32+03:00
[INFO] -----
```




Testcontainers Cloud

BY ATOMICJAR



Go to testcontainers.cloud and start testing!

Test Without Limits. Ship With Confidence.



Testcontainers
Cloud BY ATOMICJAR



Developer-first Testing

Test everything on your laptop without worrying about resources; no local docker daemon needed



Effortlessly Fast CI

Run your ever-growing test suite without scaling your CI, and speed it up by running tests in parallel



Reliable Test Suites

Enhance team efficiency by getting rid of flaky tests and ensuring consistency from dev to CI



Enterprise-ready

Adopt Testcontainers safely with VM-based isolation, in-transit encryption, and no local privileges

Local development environment with SpringBoot 3.1

```
@TestConfiguration(proxyBeanMethods = false)
public class MyContainersConfiguration {

    @Bean
    @ServiceConnection
    public Neo4jContainer<?> neo4jContainer() {
        return new Neo4jContainer<>("neo4j:5");
    }
}

public class TestMyApplication {

    public static void main(String[] args) {
        SpringApplication.from(MyApplication::main).with(MyContainersConfiguration.class).run(args);
    }
}
```


What's **next**?



– testcontainers.com



– testcontainers.cloud



– slack.testcontainers.org