

# MAVEN'S HIDDEN SECRETS AND TREASURES TO SPEED UP YOUR BUILD



adesso



#DevoxxPL



**MAVEN'S HIDDEN SECRETS AND TREASURES TO SPEED  
UP YOUR BUILD**



# GOLDEN AGE OF PIRACY





# BLACKBEARD

**Born**

Edward Teach

c. 1680

(presumed) [Bristol, England](#)

**Died**

22 November 1718

(aged 35–40)

[Ocracoke, North Carolina,  
British America](#)

**Cause of death**

[Killed in action \(stab wounds  
& gunshot wounds\)](#)

[Piratical career](#)



## ***ABOUT KO TURK***

***DEVELOPER (JAVA / KOTLIN)***

***SPEAKER***

***COMMUNITY LEAD***

***BLOGGER***

***WWW.KOTURK.NL***

# OUR JOURNEY



# TREASUREMAP

CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

PITFALLS

LIFECYCLES

SETUP MAVEN

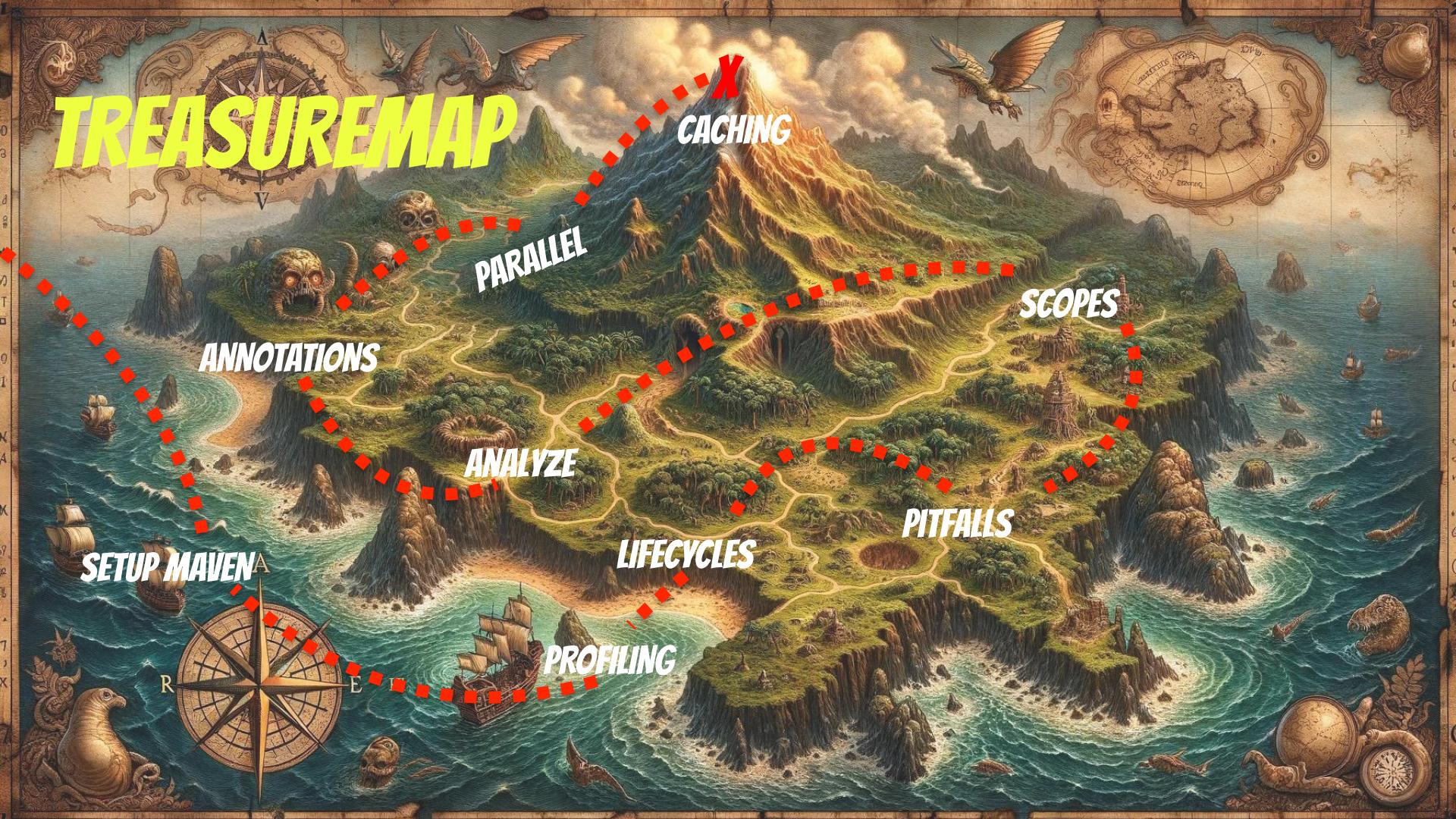
PROFILING

R E

N

S

W



# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

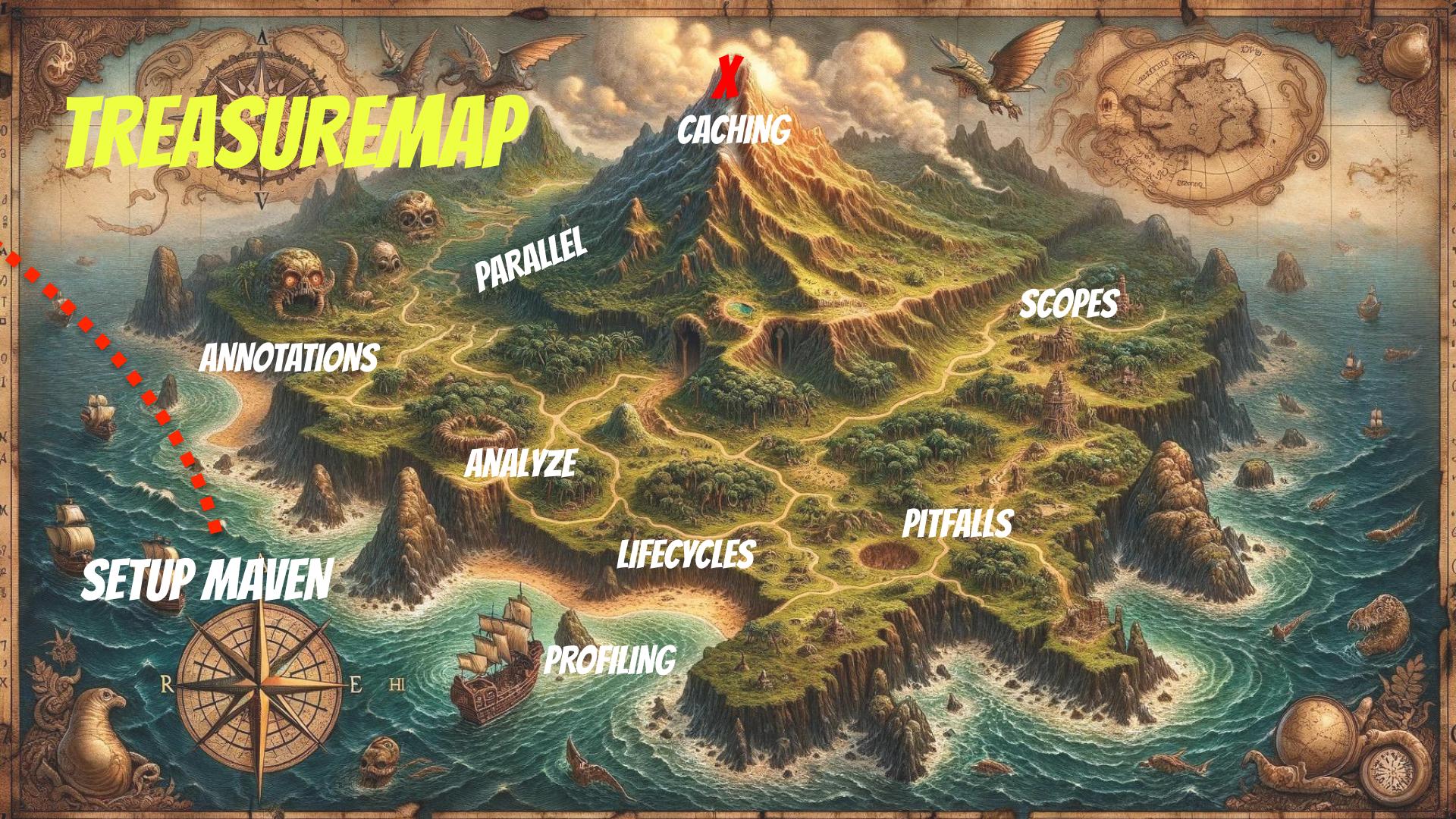
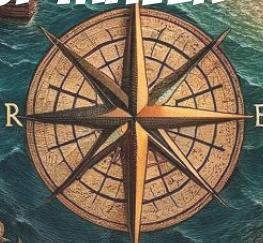
ANALYZE

PITFALLS

LIFECYCLES

PROFILING

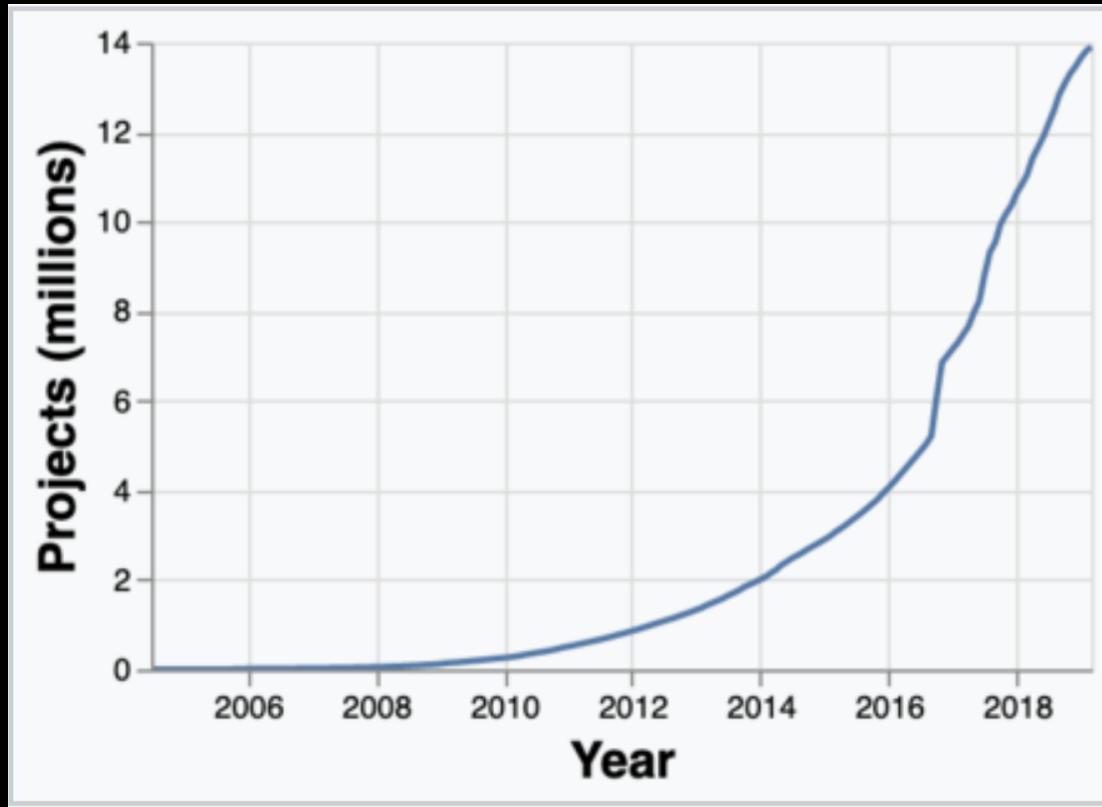
SETUP MAVEN





**WHO IS USING  
MAVEN? GRADLE? ANT?**

# **BUILDING WITH MAVEN**



# **BUILDING WITH MAVEN**

**2023: Maven 74 %, Gradle 46 %, Ant 6 %**

**2022: Maven 73 %, Gradle 50 %, Ant 8 %**

**2021: Maven 72 %, Gradle 49 %, Ant 8 %**

**2020: Maven 71 %, Gradle 48 %, Ant 9 %**

**2019: Maven 70 %, Gradle 53 %, Ant 13 %**

**2018: Maven 68 %, Gradle 47 %, Ant 11 %**

**2017: Maven 71 %, Gradle 42 %, Ant 15 %**

# ***HOW IT STARTED***



Maven was created by Jason van Zyl in 2002 and began as a sub-project of [Apache Turbine](#).

In 2003 Maven was accepted as a top level [Apache Software Foundation](#) project.

## ***OPTIONAL: DOWNLOAD MAVEN***

***DOWNLOAD VIA SITE:*** [HTTPS://MAVEN.APACHE.ORG/DOWNLOAD.CGI](https://maven.apache.org/download.cgi)

***DOWNLOAD VIA BREW:***

```
$ brew install maven
```

***DOWNLOAD VIA SDK:***

```
$ sdk install maven <version>
```

**SDKMAN!**

***OR USE A MVN WRAPPER***

***DEMO***



# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

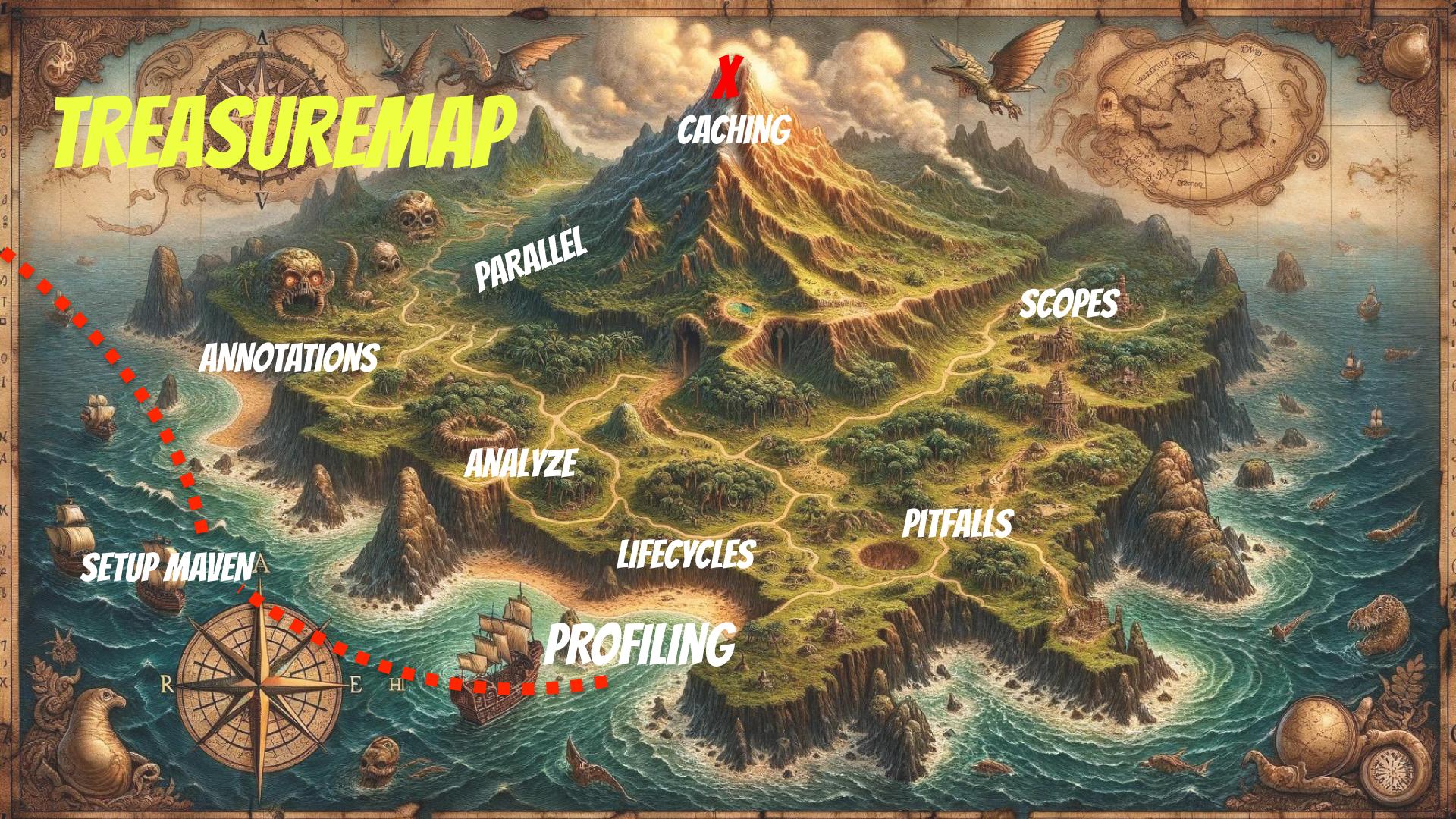
PITFALLS

LIFECYCLES

PROFILING

SETUP MAVEN

R E HI



# **PROFILING**

# **YOUR BUILD**

# **PROFILING WITH JCGAY/MAVEN-PROFILER**

Adding .mvn/extensions.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<extensions>
    <extension>
        <groupId>fr.jcgay.maven</groupId>
        <artifactId>maven-profiler</artifactId>
        <version>3.2</version>
    </extension>
</extensions>
```

```
mvn clean install -Dprofile
```

**DEMO**

**LET'S SEE HOW OUR BUILD DOES**

# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

ANALYZE

SCOPES

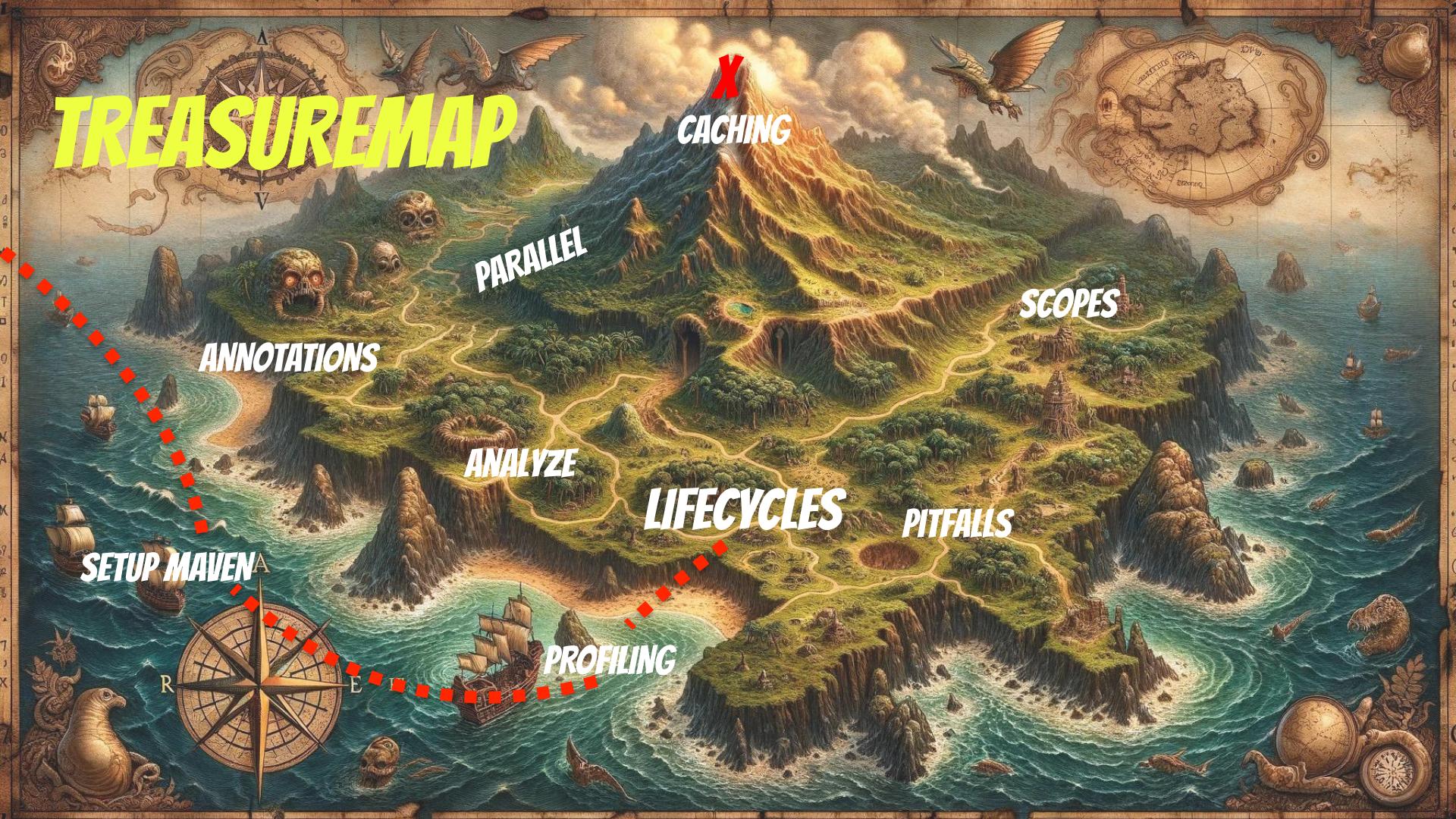
LIFECYCLES

PITFALLS

SETUP MAVENA

PROFILING

R E



# LIFECYCLES

CLEAN

DEFAULT

SITE



**CLEAN**

**PRE-CLEAN**

**CLEAN**

**POST-CLEAN**

*(CLEANS THE TARGET FOLDER)*

# ***DEMO***

# ***USING LIFE CYCLES***



**DEFAULT LIFECYCLE**

**LIST**

***THE PHASES OF BUILD  
IN CASE OF PACKAGING  
JAR***

**PROCES-RESOURCES**

*(COPY RESOURCES)*

**COMPILE**

*(COMPILES SOURCE CODE)*

**PROCES-TEST-RESOURCES**

*(COPY RESOURCES)*

**TEST-COMPILE**

*(COPY TEST RESOURCES)*

**TEST**

*(RUNS UNIT TESTS)*

**PACKAGE**

*(CREATE DISTRIBUTABLE ARTIFACT)*

**INSTALL**

*(INSTALS PACKAGE IN LOCAL REPO)*

**DEPLOY**

*(COPIES TO REMOTE REPO)*

# ***THE PHASES OF SITE***

**PRE-SITE**

**SITE**

**POST-SITE**

**SITE-DEPLOY**

*(CREATE DOCUMENTATIONS)*

*(UPLOAD TO WEB SERVER)*

**A PHASE IS BIND  
TO A PLUGIN AND GOAL**



# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

SETUP MAVEN

PITFALLS

LIFECYCLES

PROFILING

R E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

# **THE (MAVEN) PIRATE LIFECYCLES**

```
$ mvn clean
```



LIFECYCLES

**CLEAN**

PHASES

**PRE-CLEAN**

PLUGIN

**ECHO-MAVEN-PLUGIN : PRE-CLEAN**

1

\*

GOAL

**CLEAN**

**CLEAN**

:

**CLEAN**

# **ECHO-MAVEN-PLUGIN**

## **DEMO**

# **WHO'S DOING MVN CLEAN INSTALL?**



**WHAT IF I TOLD YOU**

**MVN VERIFY  
IS BETTER THAN  
MVN CLEAN INSTALL**

# WHAT GIVES PEOPLE FEELINGS OF POWER





**BY ANDRES ALMIRAY**



**[HTTPS://GITHUB.COM/AALMIRAY/MVN-CLEAN-INSTALL/](https://github.com/aalmiray/mvn-clean-install/)**

# **THE (MAVEN) PIRATE LIFECYCLES**

```
$ mvn clean install
```



**CLEAN**  
**BUILD**

# PHASES WITH PACKAGING

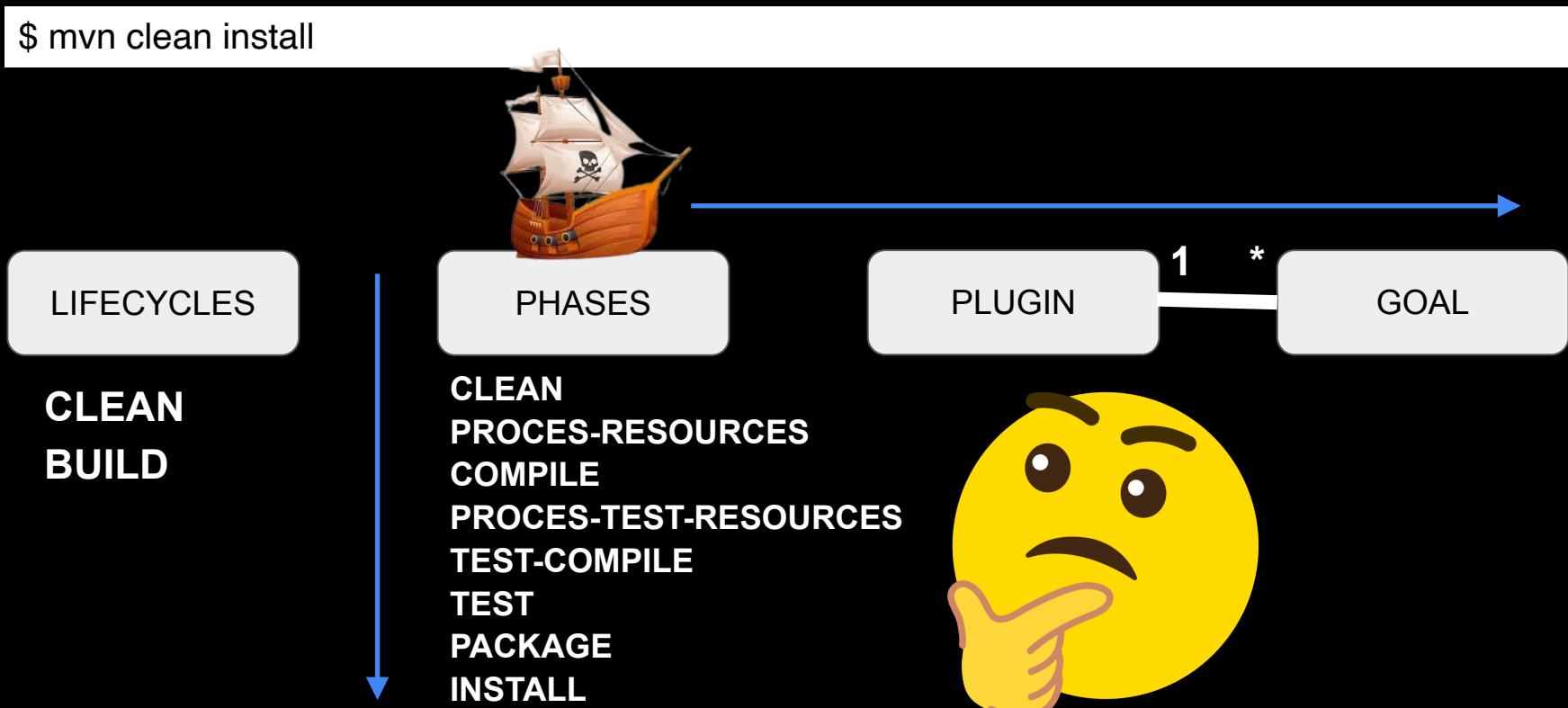
JAR



[HTTPS://MAVEN.APACHE.ORG/REF/3.9.8/MAVEN-CORE/DEFAULT-BINDINGS.HTML](https://maven.apache.org/ref/3.9.8/maven-core/default-bindings.html)

# THE (MAVEN) PIRATE LIFECYCLES

```
$ mvn clean install
```



# THE (MAVEN) PIRATE LIFECYCLES

```
$ mvn clean install
```



LIFECYCLES	PHASES	PLUGIN	GOAL
CLEAN	CLEAN	maven-clean-plugin	: clean
BUILD	PROCES-RESOURCES	maven-resources-plugin	: resources
	COMPILE	maven-compiler-plugin	: compile
	PROCES-TEST-RESOURCES	maven-resources-plugin	: testResources
	TEST-COMPILE	maven-compiler-plugin	: testCompile
	TEST	maven-surefire-plugin	: test
	PACKAGE	maven-jar-plugin	: jar
	INSTALL	maven-install-plugin	: install

# **MVN VERIFY**

# **DEMO**

# **THE (MAVEN) PIRATE LIFECYCLES**

```
$ mvn site-deploy
```



LIFECYCLES

SITE

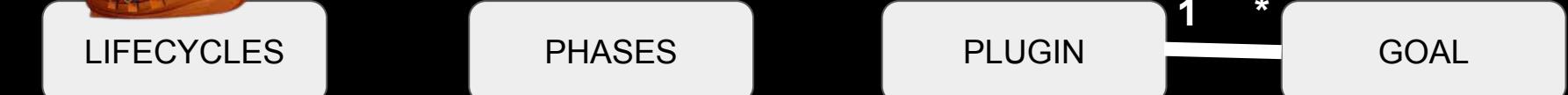
PHASES

PLUGIN

GOAL

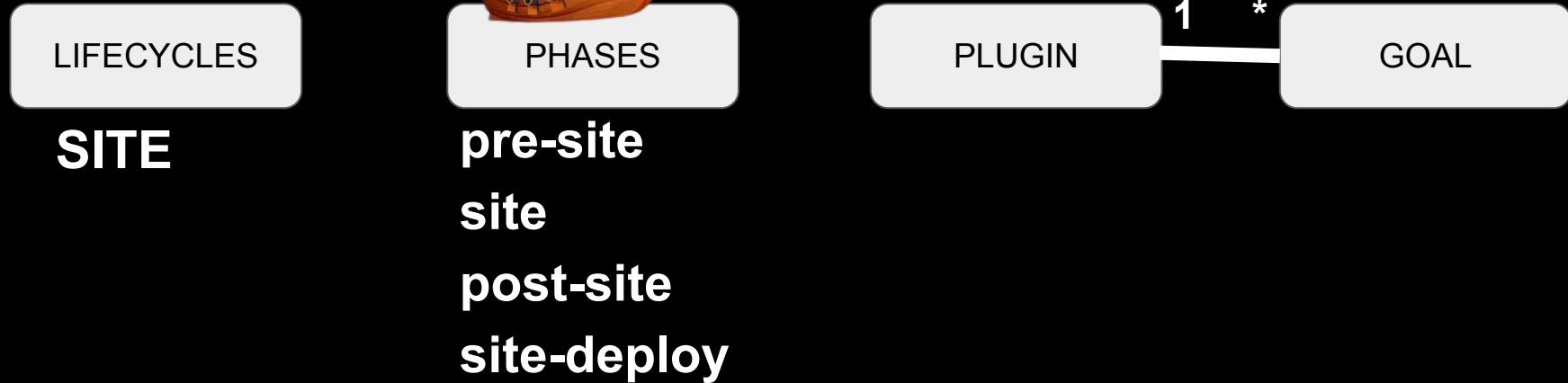
1

\*



# **THE (MAVEN) PIRATE LIFECYCLES**

```
$ mvn site-deploy
```



# **THE (MAVEN) PIRATE LIFECYCLES**

```
$ mvn site-deploy
```



LIFECYCLES

SITE

PHASES

pre-site  
site  
post-site  
site-deploy

PLUGIN

site : site

1

\*

GOAL

:

site

:

deploy

# ***BEWARE, YOU ARE NOT INTEGRATION TESTING!***

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-failsafe-plugin</artifactId>
    <version>2.22.1</version>
    <executions>
        <execution>
            <id>default-system-test</id>
            <goals>
                <goal>integration-test</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

# TREASUREMAP

X  
CACHING

ANNOTATIONS

PARALLEL

SCOPES

SETUP MAVEN

ANALYZE

PITFALLS

LIFECYCLES

PROFILING

**CHANGE YOUR  
SCOPES!**

**CHANGE YOUR  
SCOPES! → DEMO**

# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

SETUP MAVEN

PITFALLS

LIFECYCLES

PROFILING

R E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

W E

N S

S N

E W

# ***DELETING UNUSED DEPENDENCIES***

```
$ mvn dependency:analyze
```

## **OUTPUT:**

```
[WARNING] Unused declared dependencies found:  
[WARNING]   org.springframework.boot:spring-boot-starter-actuator  
compile
```

***DEMO***

# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

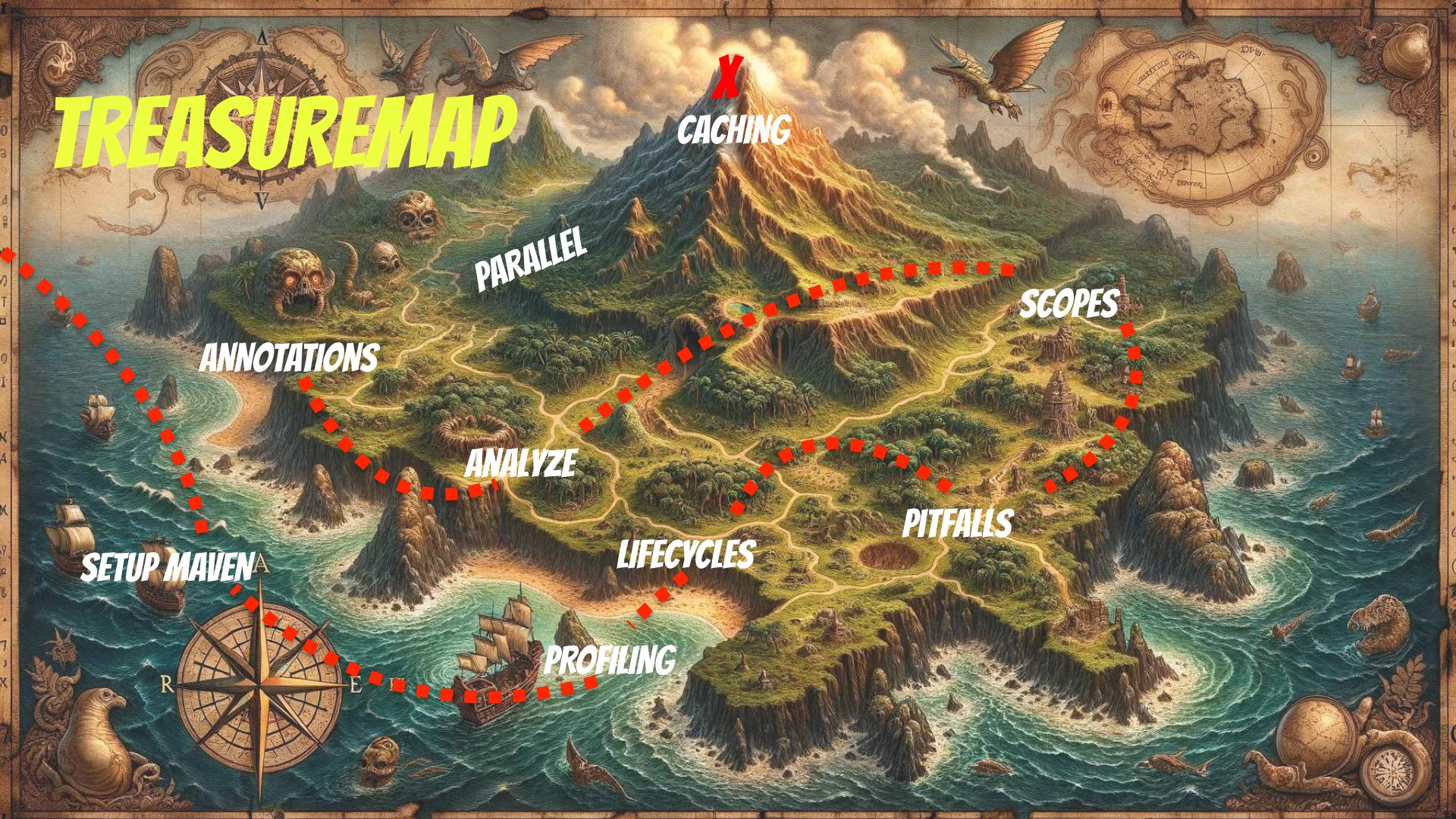
PITFALLS

LIFECYCLES

SETUP MAVEN

PROFILING

R E





# **TALK WITH KARL HEINZ MARBAISE**

# ***TESTING THE RIGHT SLICES OF YOUR APPLICATION***

**@SpringBootTest**

**@WebMvcTest**

**@JsonTest**

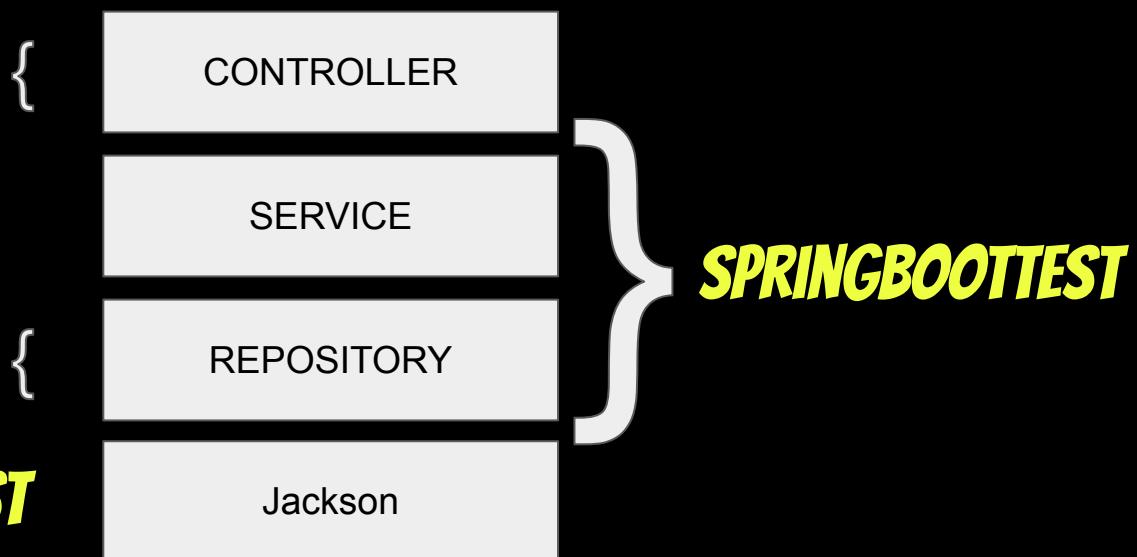
**@DataJPATest**

**@RestClientTests**

# ***TESTING THE RIGHT SLICES OF YOUR APPLICATION***

**WEBMVCTEST**

*WEBFLUXTEST*



**DATAJPATEST**

*JDBCTEST*

*DATAMONGOTEST*

*DATAREDISTEST*

**JSONTEST**



## Test slice annotations

- ✓ `@WebMvcTest`
- ✓ `@RestClientTest`
- ✓ `@DataJpaTest`
- ✓ `@GraphQLTest`
- ✓ `@WebServerServerTest`
- ✓ `@WebServiceClientTest`

<https://docs.spring.io/spring-boot/docs/current/reference/html/test-auto-configuration.html>

<https://spring.io/blog/2016/08/30/custom-test-slice-with-spring-boot-1-4>



<https://youtu.be/5Z5sa9HpQrE?si=JjIR-DfcEgB7TAg9&t=630>

# TREASUREMAP

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

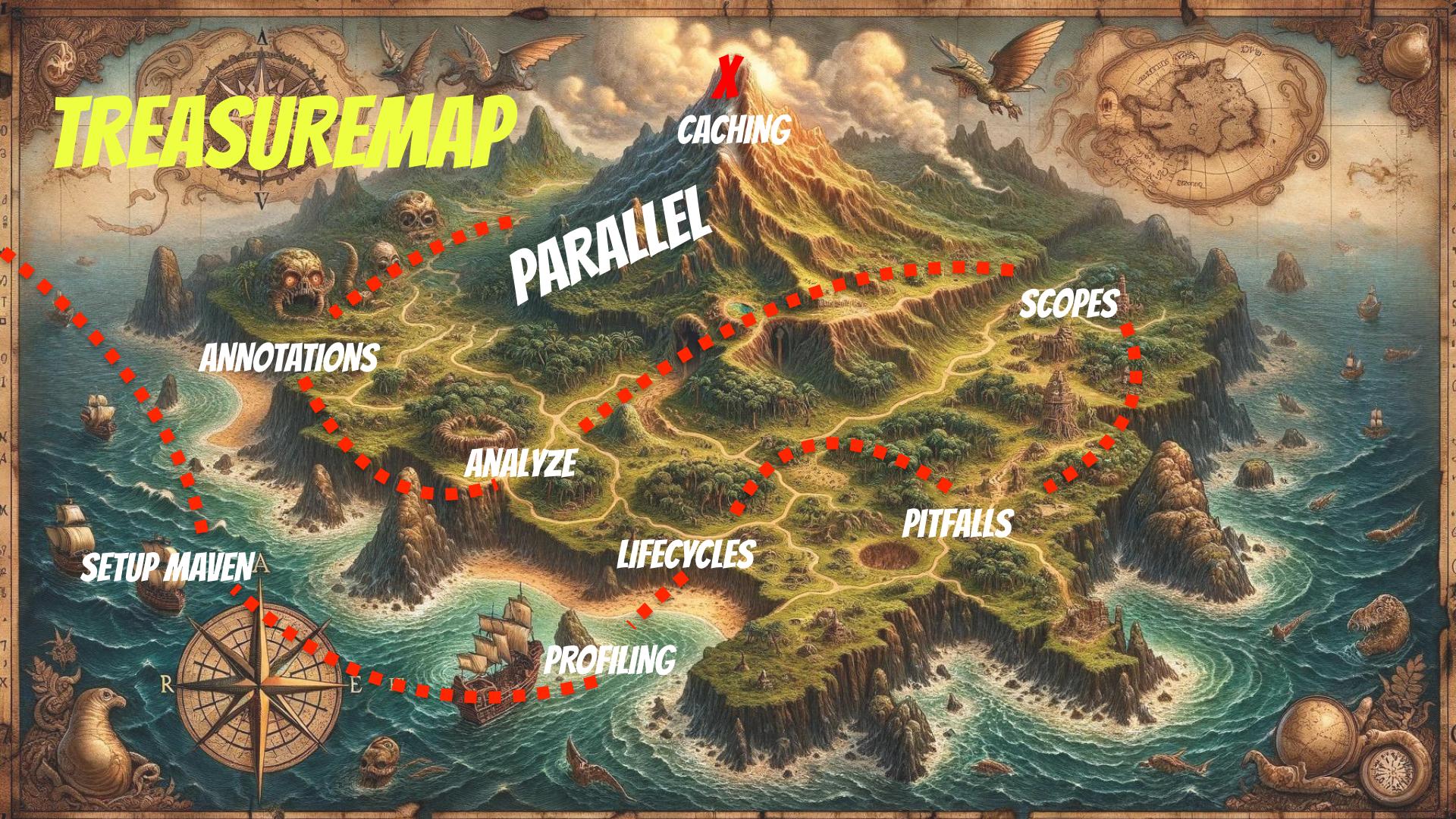
PITFALLS

LIFECYCLES

SETUP MAVEN

PROFILING

R E



**USING MULTIPLE CORES  
WHEN BUILDING**

# **USING MULTIPLE THREADS**

If your computer has only 1 CORE, use:

```
$ mvn -T1C clean install
```

If your computer has more, like my macbook has 12 COREs, use:

```
$ mvn -T12C clean install
```

***IF SOMEONE DIDN'T LOCK THEIR LAPTOP***

***.MVN/MAVEN.CONFIG***

**TO BE ADDED BY OPENREWRITE**

[HTTPS://GITHUB.COM/OPENREWRITE/  
REWRITE/ISSUES/4177](https://github.com/openrewrite/rewrite/issues/4177)

# TREASUREMAP

CACHING

PARALLEL

ANNOTATIONS

ANALYZE

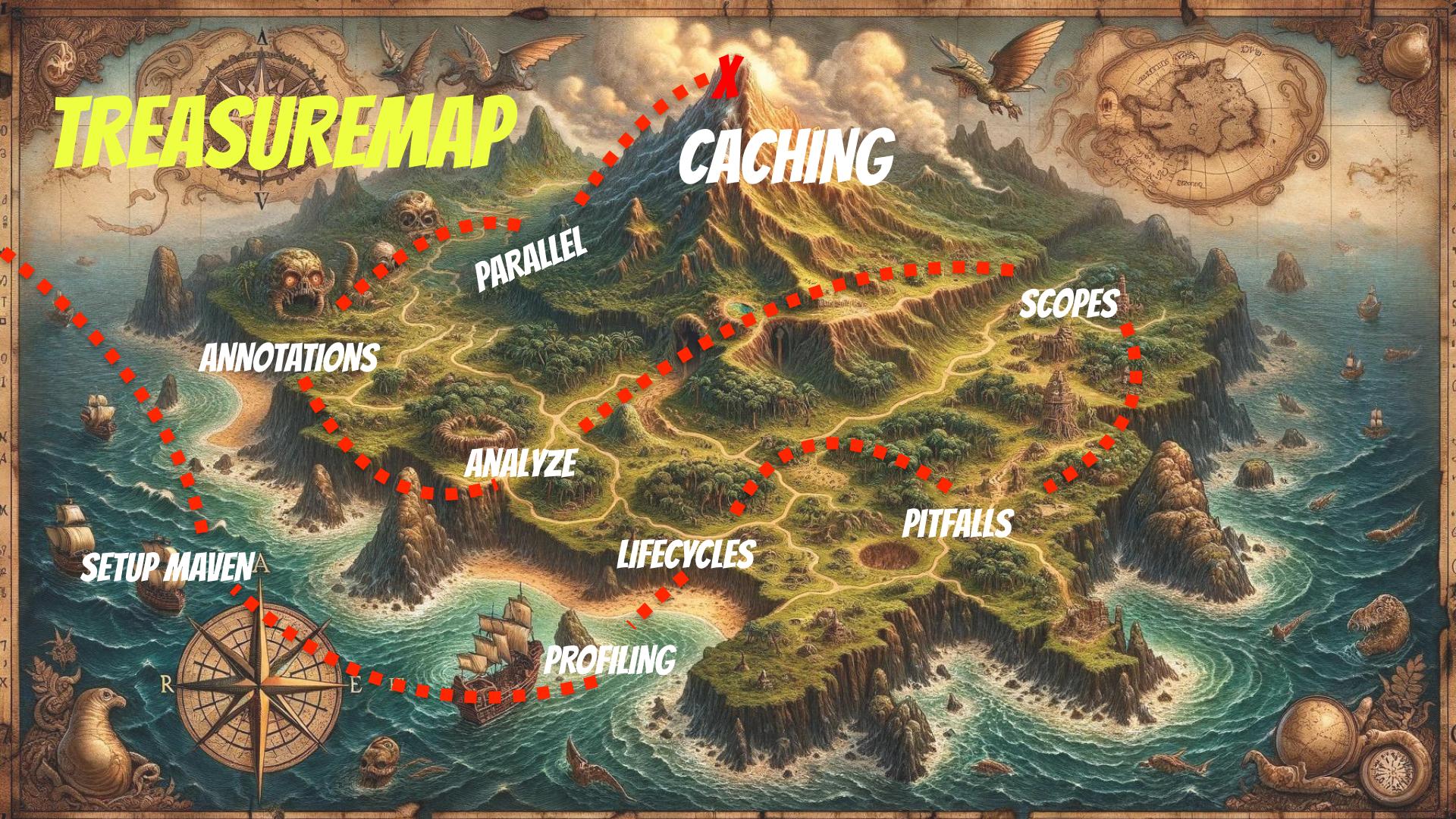
SETUP MAVEN

LIFECYCLES

PROFILING

SCOPES

PITFALLS



**CACHING ARTIFACTS**

**→ LOCAL**

# **ADJUST YOUR POM FILE**

```
<build>
  <extensions>
    <extension>
      <groupId>org.apache.maven.extensions</groupId>
      <artifactId>maven-build-cache-extension</artifactId>
      <version>1.1.0</version>
    </extension>
  </extensions>
</build>
```

# **OR ADJUST YOUR GRADLE FILE**

```
buildCache
{
    local {
        directory = File(rootDir, "build-cache")
        removeUnusedEntriesAfterDays = 30
    }
}
```

# **WHERE DOES IT SAVE IT CACHE?**

Show the files within the build cache directory:

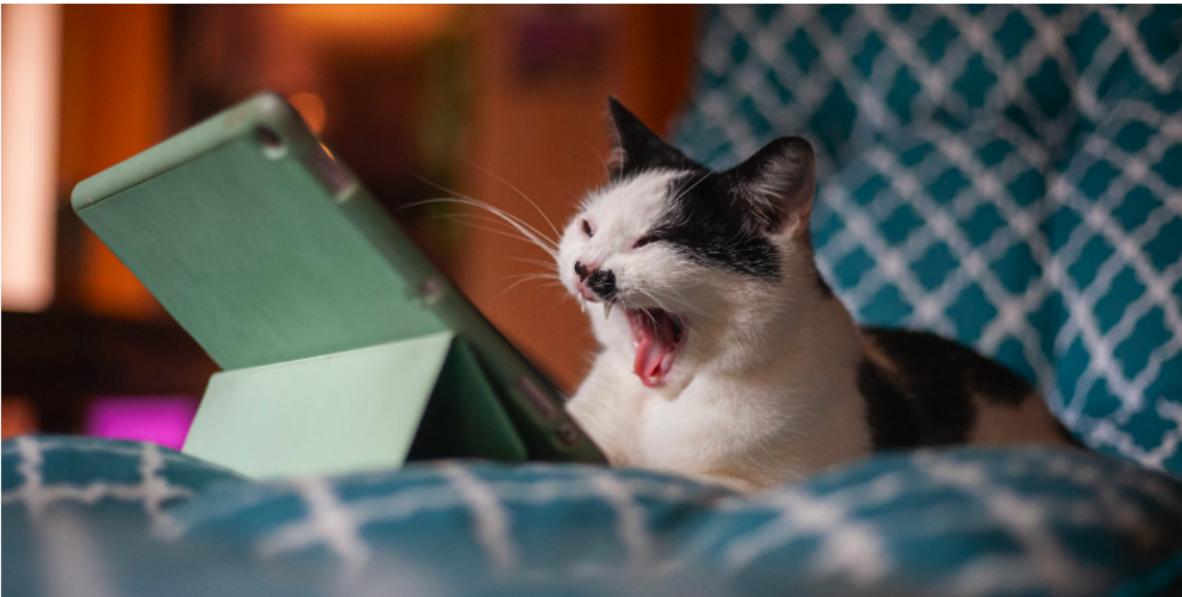
```
$ ls ~/.m2/build-cache/v1
```

com.baeldung

com.baeldung.daomodule

com.baeldung.....

## Meow moments, minimize the wait for your build to be completed!



Build caching helps you reduce your local or remote build time! Building a project with Maven or Gradle will save your artifacts to a (remote) cache. The second build will get the artifacts from the cache if nothing changes. It sounds great, right? But why was nobody aware of the feature when I asked about it during our presentation at JFall? Only two people in the room (of 100) knew what it was. This is an excellent opportunity to tell you more about this feature. So say goodbye to your extra coffee breaks, and let's go back to work and code!



**CACHING ARTIFACTS**

→ **REMOTE**

# GITHUB WORKFLOW

[HTTPS://GITHUB.COM/MARKETPLACE/ACTIONS/MAVEN-CACHE](https://github.com/marketplace/actions/maven-cache)

# TREASURES

X  
CACHING

PARALLEL

ANNOTATIONS

SCOPES

ANALYZE

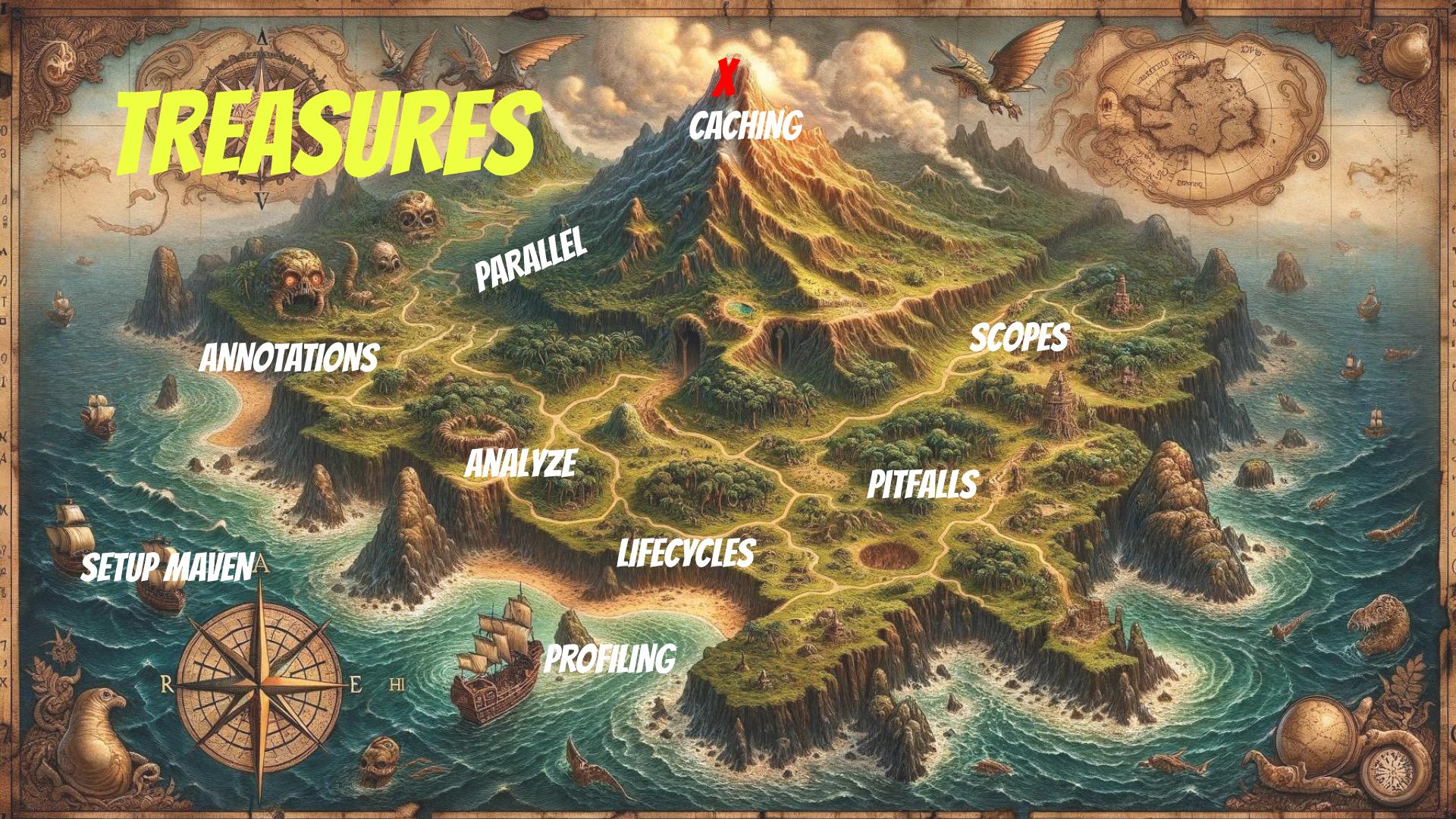
PITFALLS

LIFECYCLES

SETUP MAVENA

PROFILING

R E H I



**DEMO**

**DID WE IMPROVED OUR BUILD TIME?**

A dark, moody background featuring a stack of shiny gold coins and a single gold ring, all resting on a dark, textured surface. The lighting is dramatic, highlighting the metallic textures.

**THANK YOU**

**DON'T FORGET TO RATE MY TALK**