OPEN SOURCE CODE INSPECTION AND TESTING TOOLS

OPENSOUTHCODE
MÁLAGA – 2 DE JUNIO DE 2018

JORGE HIDALGO
ACCENTURE TECHNOLOGY
GLOBAL JAVA LEAD

MÁLAGAJUG CO-LEAD
MÁLAGA SCALA DEVELOPERS CO-LEAD

WHO I AM

Jorge Hidalgo deors deors

Global Java Lead

Custom Engineering & Architecture Lead – Accenture Spain Advanced Technology Center

Co-lead – MálagaJUG & Málaga Scala Developers

Father of two children, husband, whistle player, video gamer, sci-fi 'junkie', Star Wars 'addict', Lego brick 'wielder', Raspberry Pi fan... LLAP!

https://deors.wordpress.com
https://www.meetup.com/es-ES/MalagaJUG/
https://www.meetup.com/es-ES/Malaga-Scala/

Copyright 2018 Accenture. All rights reserved.

OPEN SOURCE POWER TOOLS

CODE INSPECTION

CODE COVERAGE

MUTATION TESTING

MOCKS, STUBS, DOUBLES

SECURITY TESTING

Copyright 2018 Accenture. All rights reserved.

MOTIVATION – WHY USE TOOLS?

QUALITY

Software craftmanship

No blaming

No last minute fixes

Client satisfaction

Boss satisfaction

Pay rise!

PRODUCTIVITY

No boring, repetitive tasks

Focus on the cool stuff

Do more in less time

Client satisfaction

Boss satisfaction

Pay rise!

PREDICTABILITY

SE as a precision work

Always on schedule

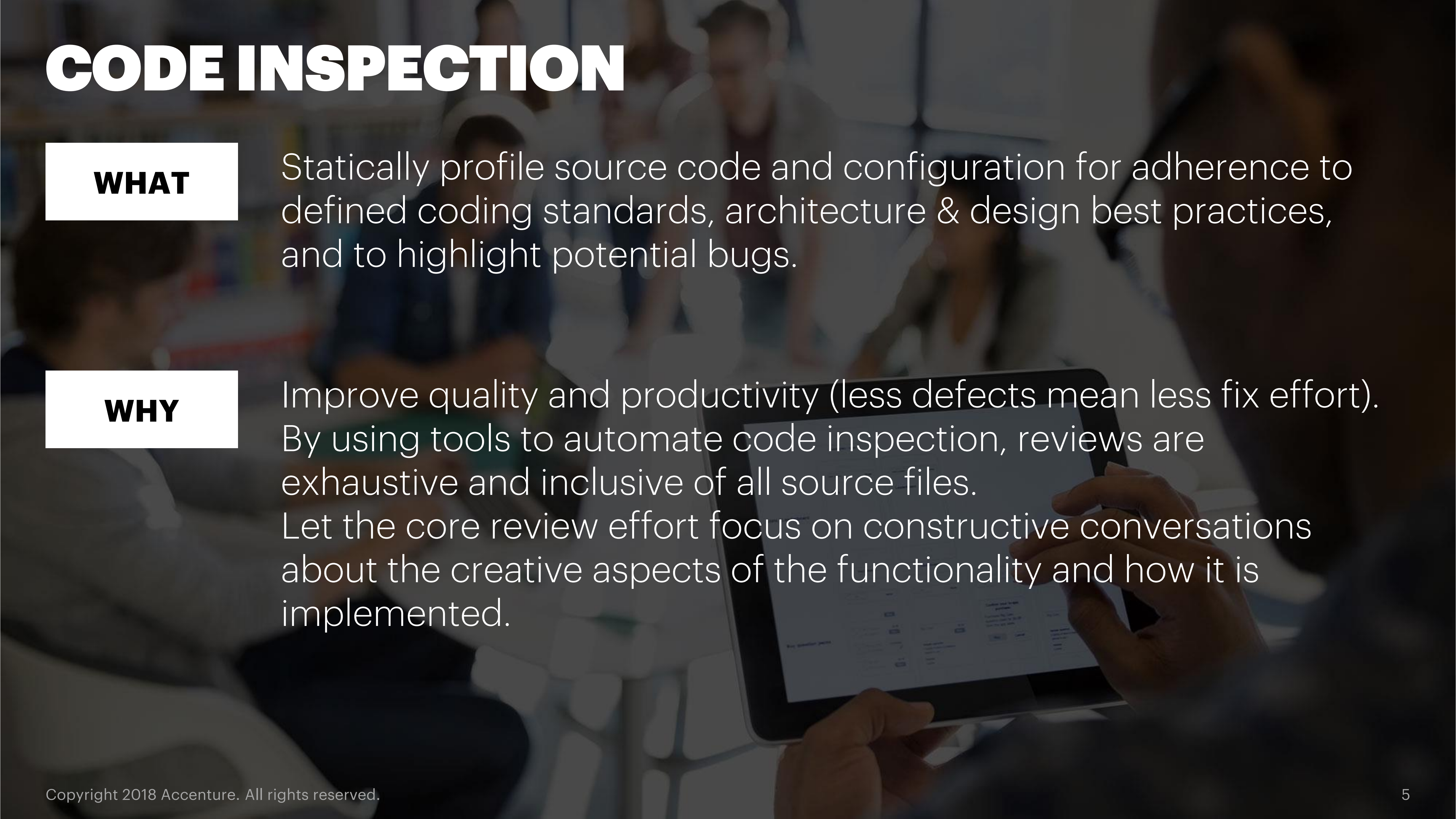
No surprises

Client satisfaction

Boss satisfaction

Pay rise!

Copyright 2018 Accenture. All rights reserved.



CODE INSPECTION

WHAT

Statically profile source code and configuration for adherence to defined coding standards, architecture & design best practices, and to highlight potential bugs.

WHY

Improve quality and productivity (less defects mean less fix effort).
By using tools to automate code inspection, reviews are exhaustive and inclusive of all source files.
Let the core review effort focus on constructive conversations about the creative aspects of the functionality and how it is implemented.

Copyright 2018 Accenture. All rights reserved.

CODE INSPECTION TOOLS

Java checkstyle FindBugs SpotBugs pmd DON'T SHOOT THE MESSENGER sonarqube IJ

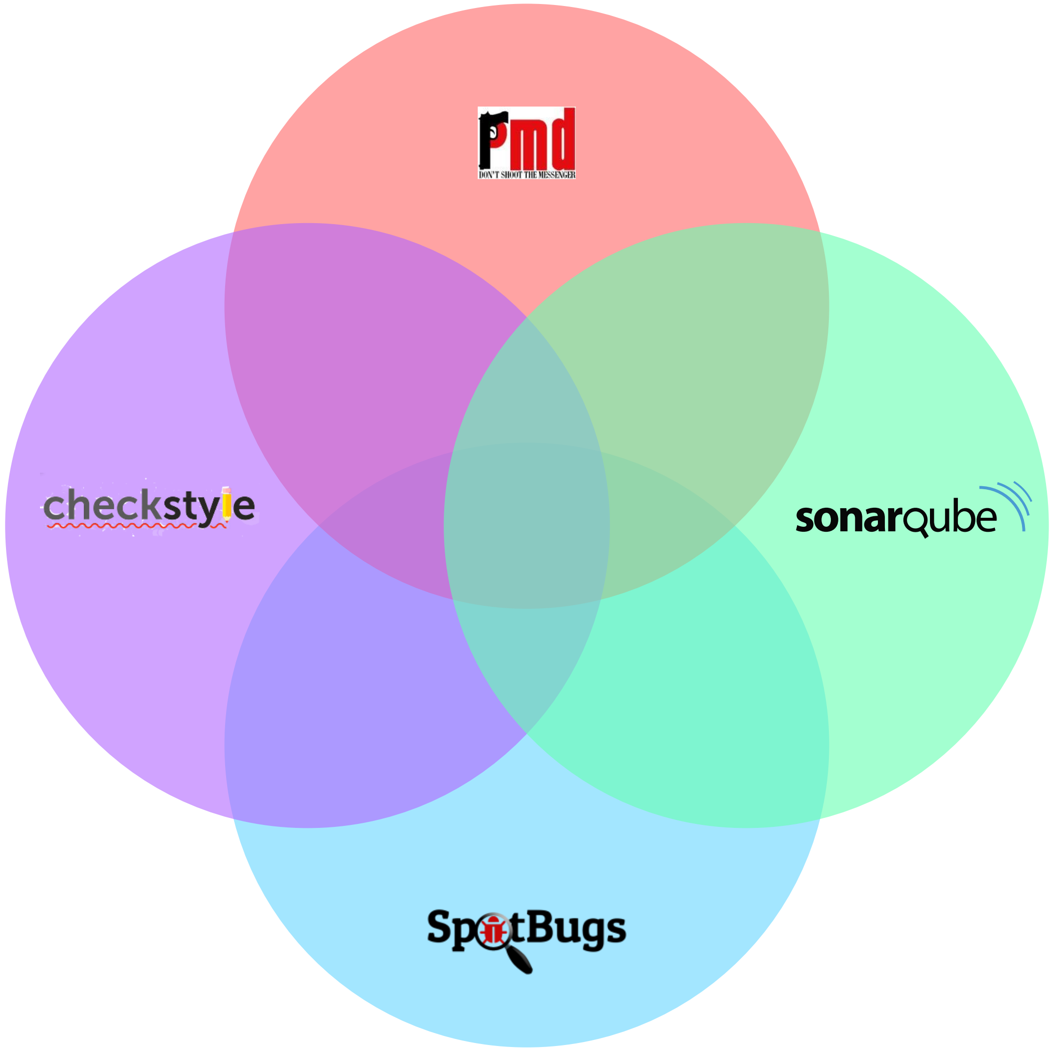
Groovy CODENARC Less Bugs Better Code sonarqube

Scala Scalastyle Scapegoat

JS TS ESLint TSLint sonarqube

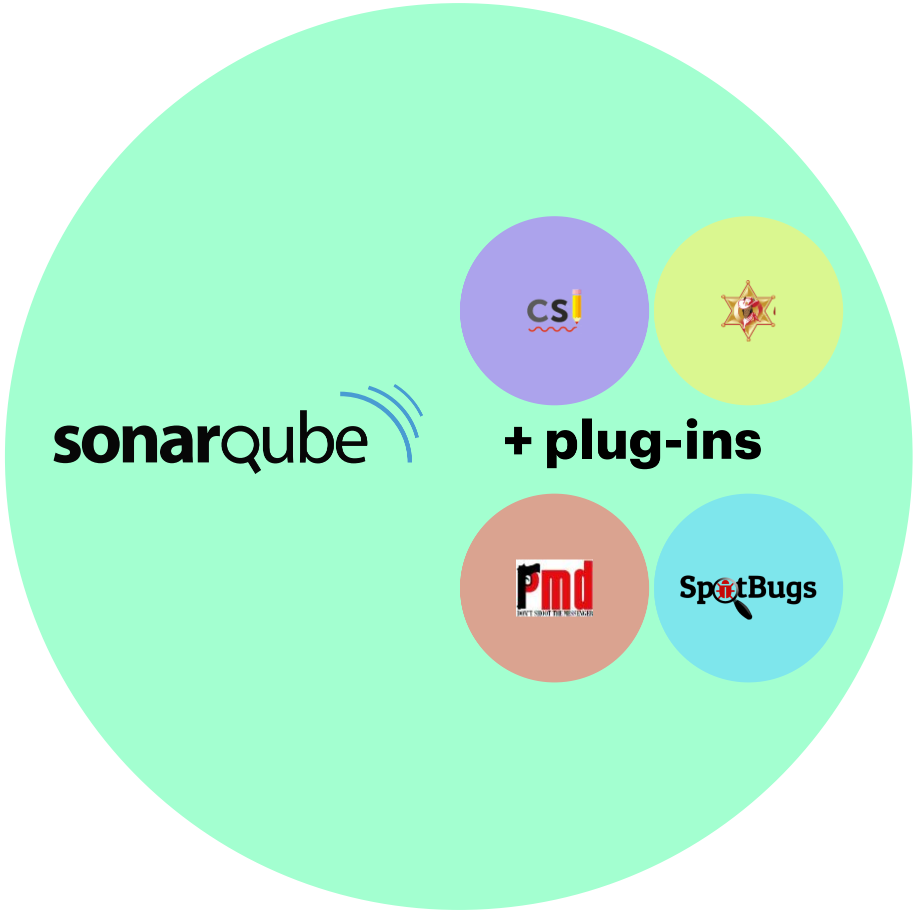
CODE INSPECTION TOOLS

Beware of overlapping (equivalent) rules!



Copyright 2018 Accenture. All rights reserved.

CODE INSPECTION TOOLS



Get the best from each of them

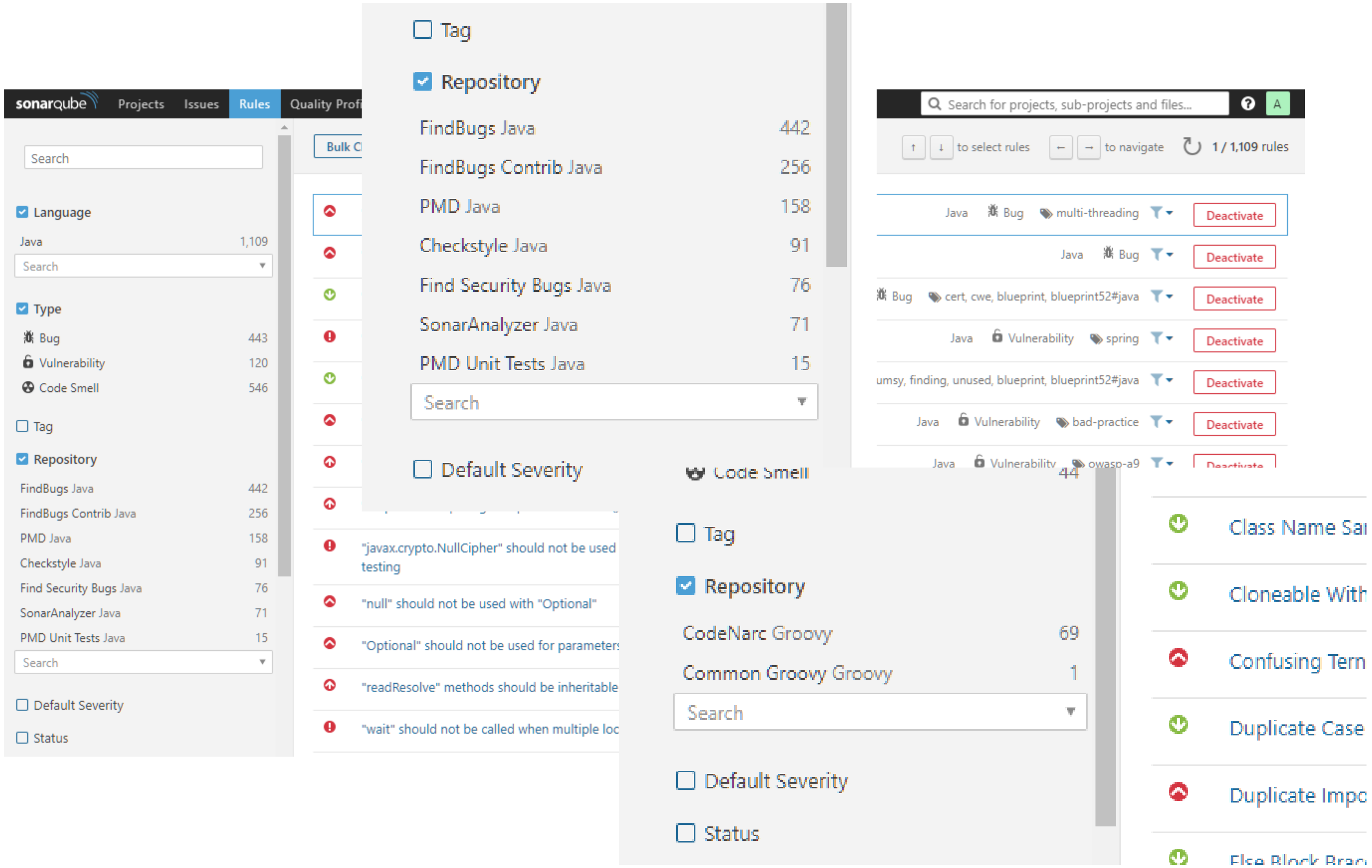
Combine outputs into a single report

Code reviews

Action plans

Copyright 2018 Accenture. All rights reserved.

CODE INSPECTION TOOLS



sonarqube Projects Issues Rules Quality Profi

Search

Language

Java	1,109

Search

Type

Bug	443
Vulnerability	120
Code Smell	546

Tag

Repository

FindBugs Java	442
FindBugs Contrib Java	256
PMD Java	158
Checkstyle Java	91
Find Security Bugs Java	76
SonarAnalyzer Java	71
PMD Unit Tests Java	15

Search

Default Severity

Status

Bulk C

"javax.crypto.NullCipher" should not be used testing

"null" should not be used with "Optional"

"Optional" should not be used for parameters

"readResolve" methods should be inheritable

"wait" should not be called when multiple loc

Search for projects, sub-projects and files...

↑ ↓ to select rules ← → to navigate 1 / 1,109 rules

Java	Bug	multi-threading	Deactivate
Java	Bug		Deactivate
	Bug	cert, cwe, blueprint, blueprint52#java	Deactivate
Java	Vulnerability	spring	Deactivate
		umsy, finding, unused, blueprint, blueprint52#java	Deactivate
Java	Vulnerability	bad-practice	Deactivate
Java	Vulnerability	owasp-a9	Deactivate

Code Smell 44

Tag

Repository

CodeNarc Groovy	69
Common Groovy Groovy	1

Search

Default Severity

Status

Class Name Sa

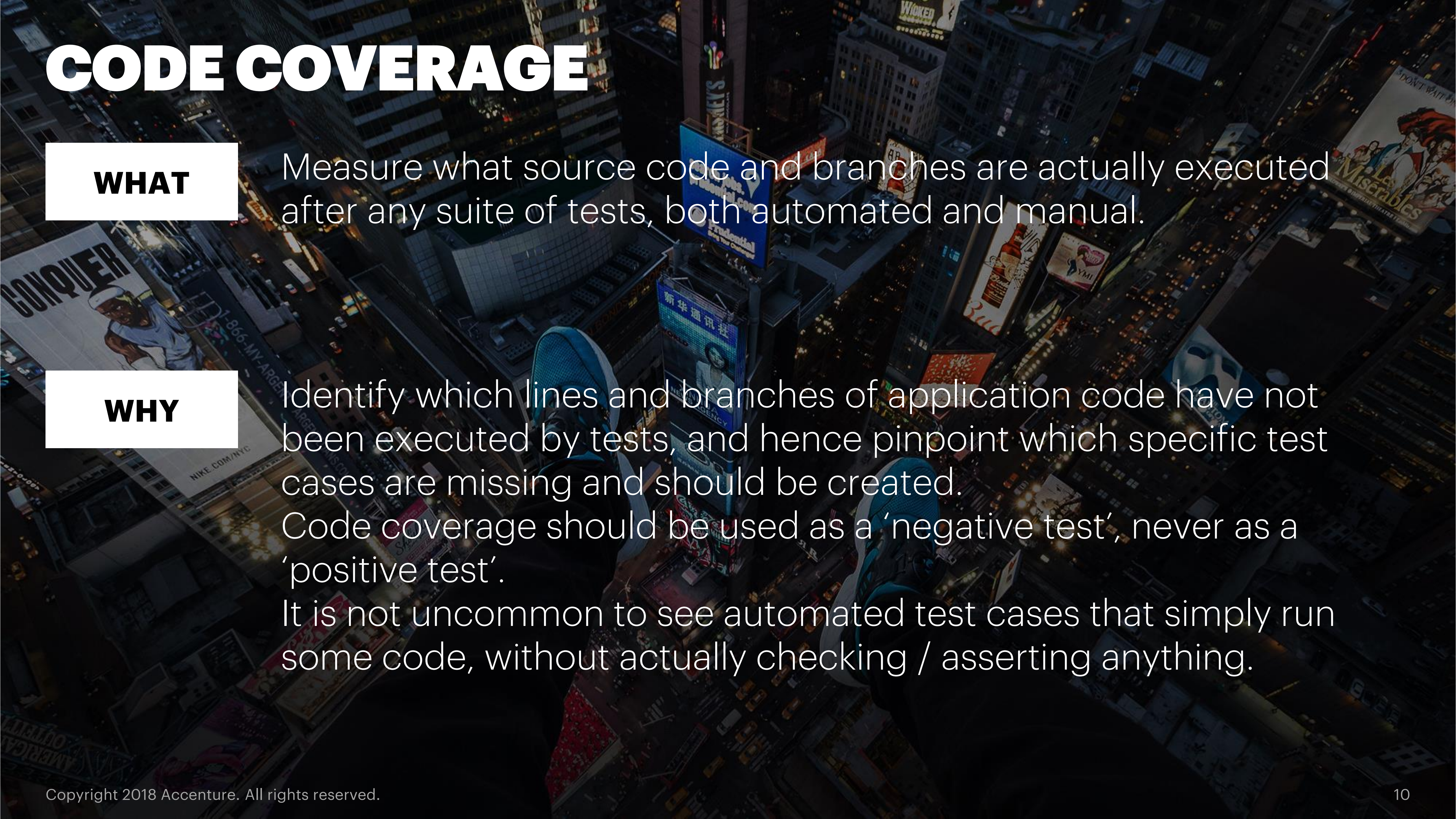
Cloneable With

Confusing Tern

Duplicate Case

Duplicate Impo

Else Block Brac



CODE COVERAGE

WHAT

Measure what source code and branches are actually executed after any suite of tests, both automated and manual.

WHY

Identify which lines and branches of application code have not been executed by tests, and hence pinpoint which specific test cases are missing and should be created.
Code coverage should be used as a ‘negative test’, never as a ‘positive test’.
It is not uncommon to see automated test cases that simply run some code, without actually checking / asserting anything.

Copyright 2018 Accenture. All rights reserved.

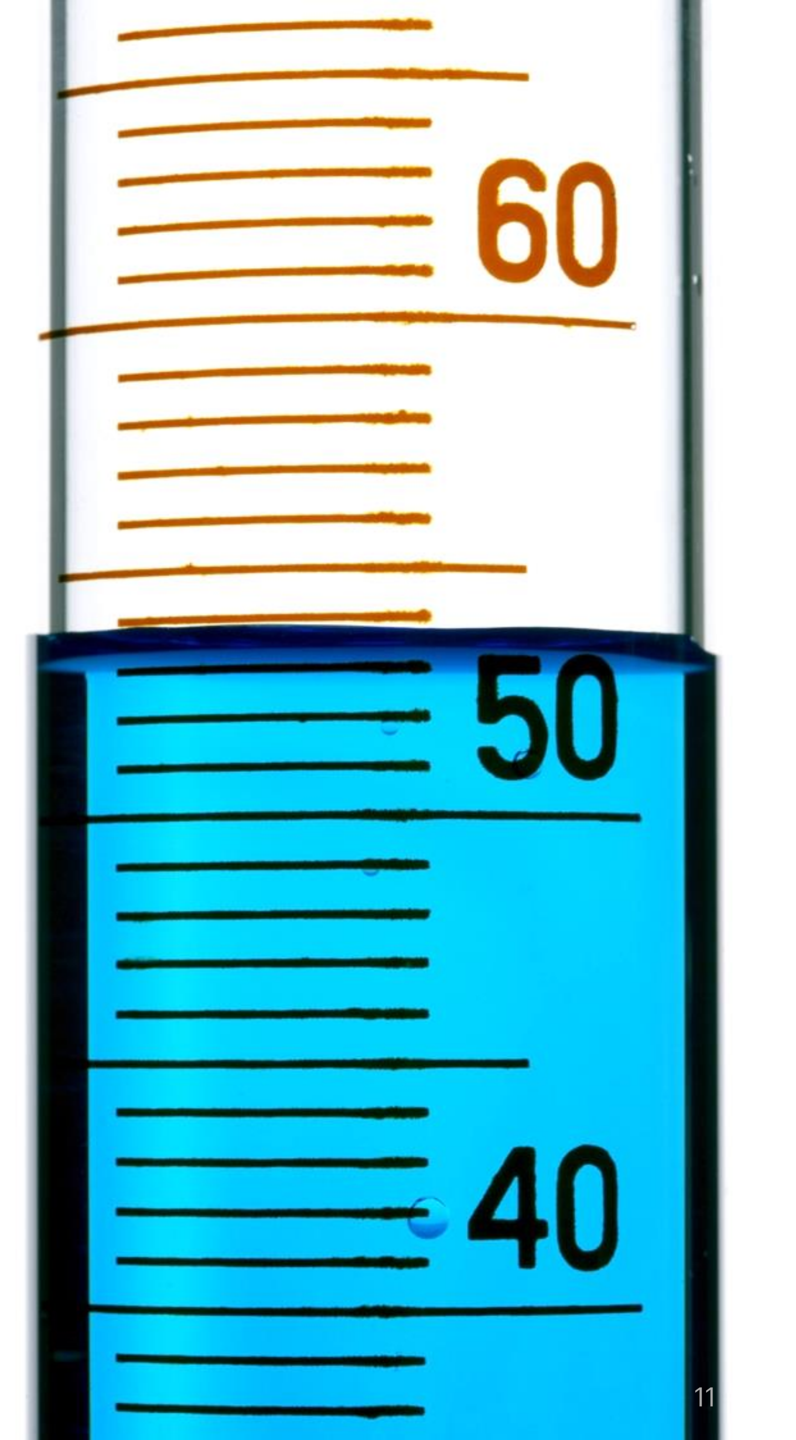
CODE COVERAGE TOOLS

Java — JACOCO Java Code Coverage — Clover — eclipse {ECLEMMA} — JCov — Cobertura

Groovy — JACOCO Java Code Coverage — Clover

Scala — SCOVERAGE SCALA CODE COVERAGE

JS TS — istanbul — isparta



Copyright 2018 Accenture. All rights reserved.

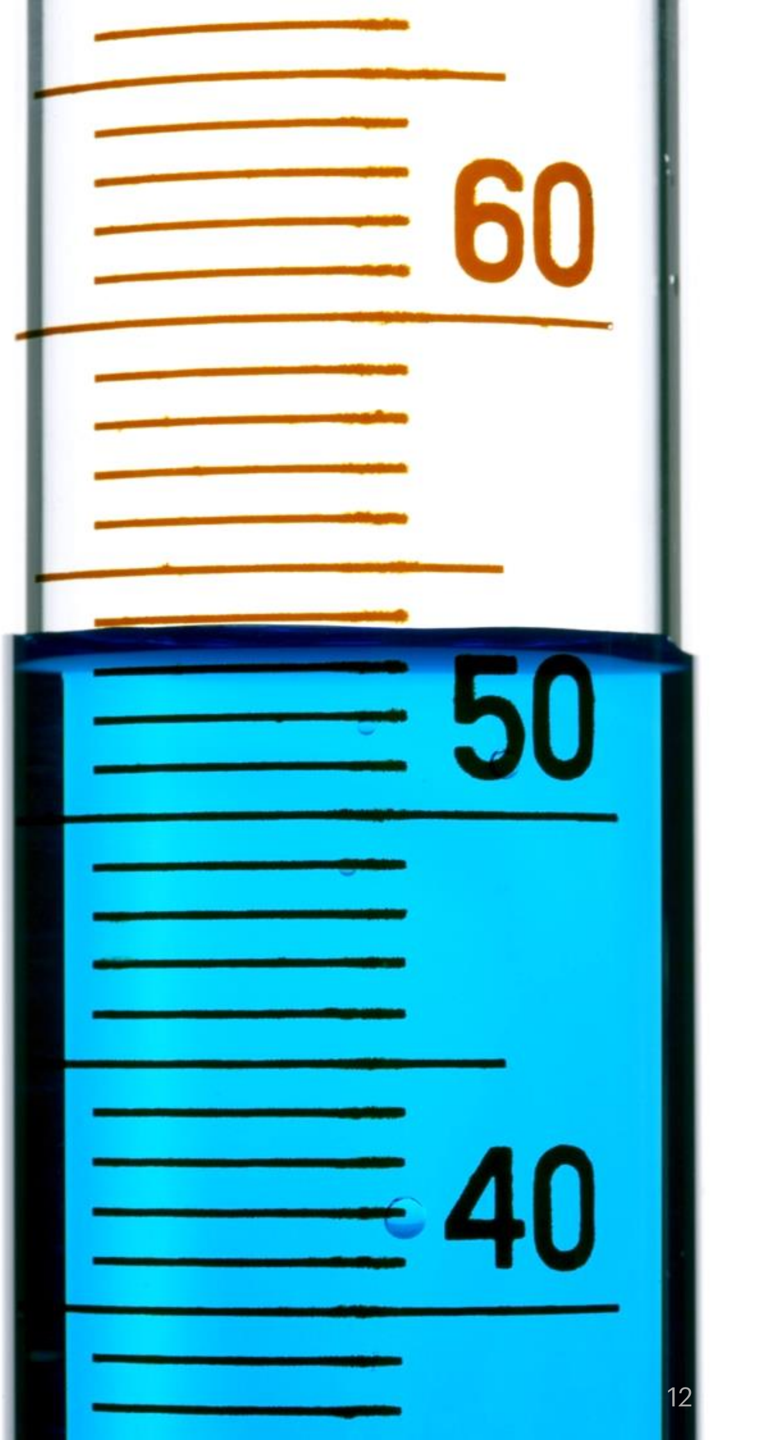
CODE COVERAGE TOOLS

JACOCO
Java Code Coverage

Use it to gather coverage from manual tests, if you don’t have automated

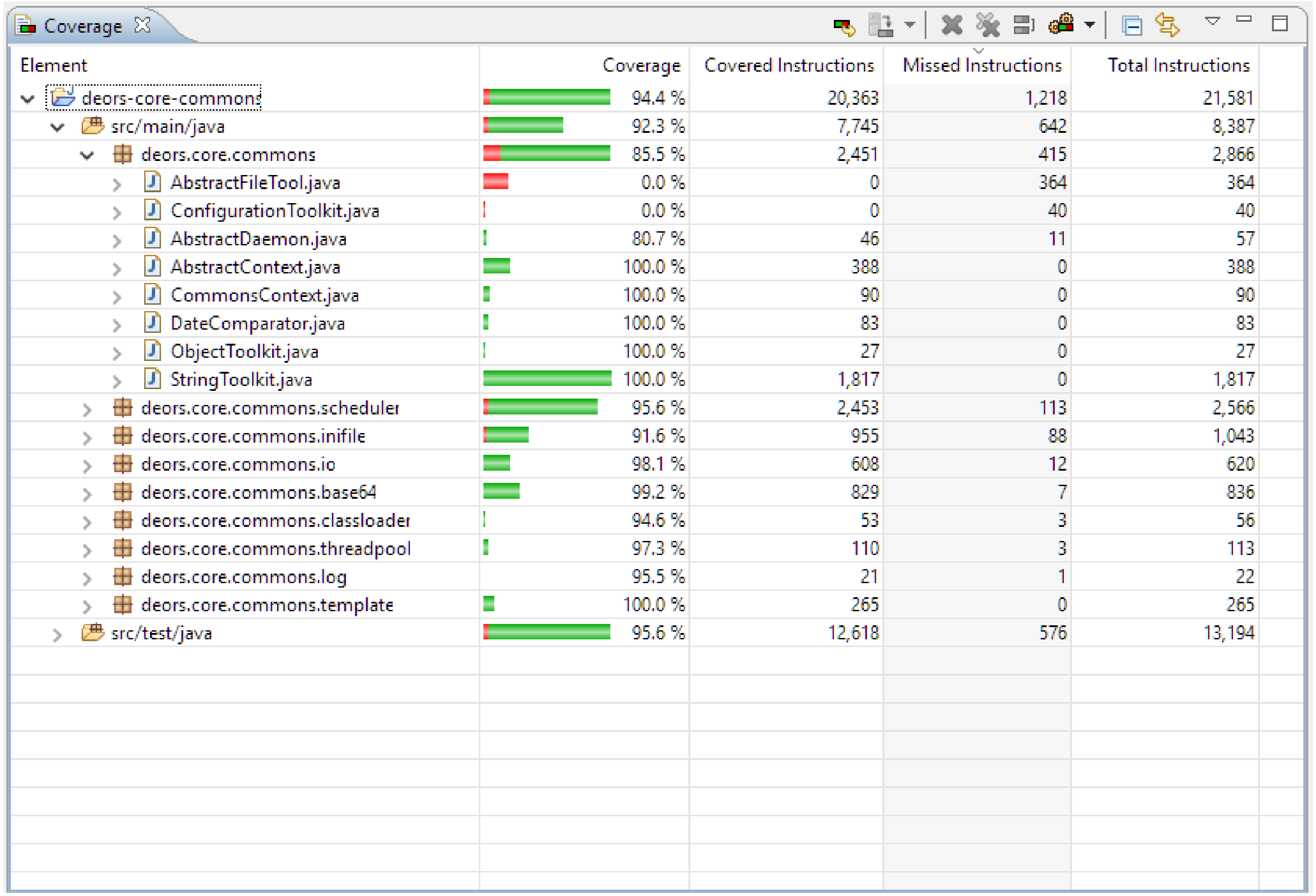
Use EclEmma inside Eclipse to ensure all key test cases are covered by tests

Combine with SonarQube listener to get metrics per every single test executed

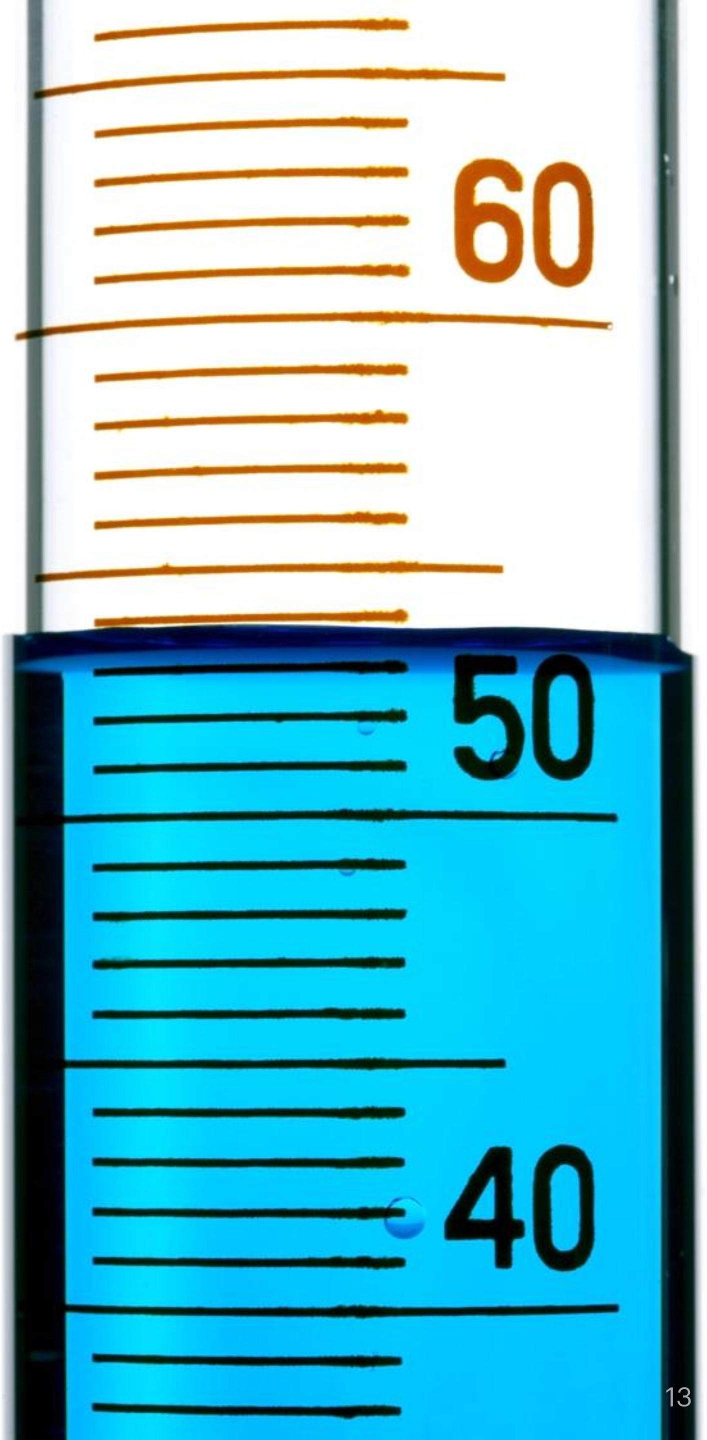


Copyright 2018 Accenture. All rights reserved.

CODE COVERAGE TOOLS



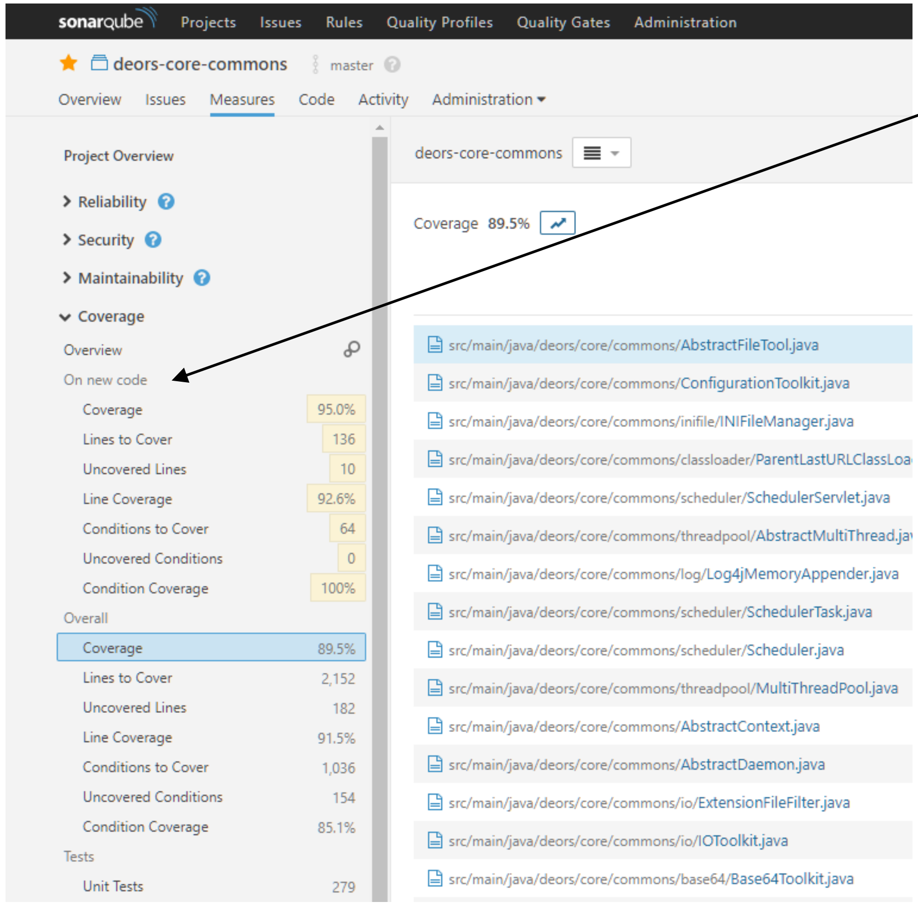
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
deors-core-commons	94.4 %	20,363	1,218	21,581
src/main/java	92.3 %	7,745	642	8,387
deors.core.commons	85.5 %	2,451	415	2,866
AbstractFileTool.java	0.0 %	0	364	364
ConfigurationToolkit.java	0.0 %	0	40	40
AbstractDaemon.java	80.7 %	46	11	57
AbstractContext.java	100.0 %	388	0	388
CommonsContext.java	100.0 %	90	0	90
DateComparator.java	100.0 %	83	0	83
ObjectToolkit.java	100.0 %	27	0	27
StringToolkit.java	100.0 %	1,817	0	1,817
deors.core.commons.scheduler	95.6 %	2,453	113	2,566
deors.core.commons.inifile	91.6 %	955	88	1,043
deors.core.commons.io	98.1 %	608	12	620
deors.core.commons.base64	99.2 %	829	7	836
deors.core.commons.classloader	94.6 %	53	3	56
deors.core.commons.threadpool	97.3 %	110	3	113
deors.core.commons.log	95.5 %	21	1	22
deors.core.commons.template	100.0 %	265	0	265
src/test/java	95.6 %	12,618	576	13,194



Copyright 2018 Accenture. All rights reserved.

CODE COVERAGE TOOLS

Code coverage in SonarQube:

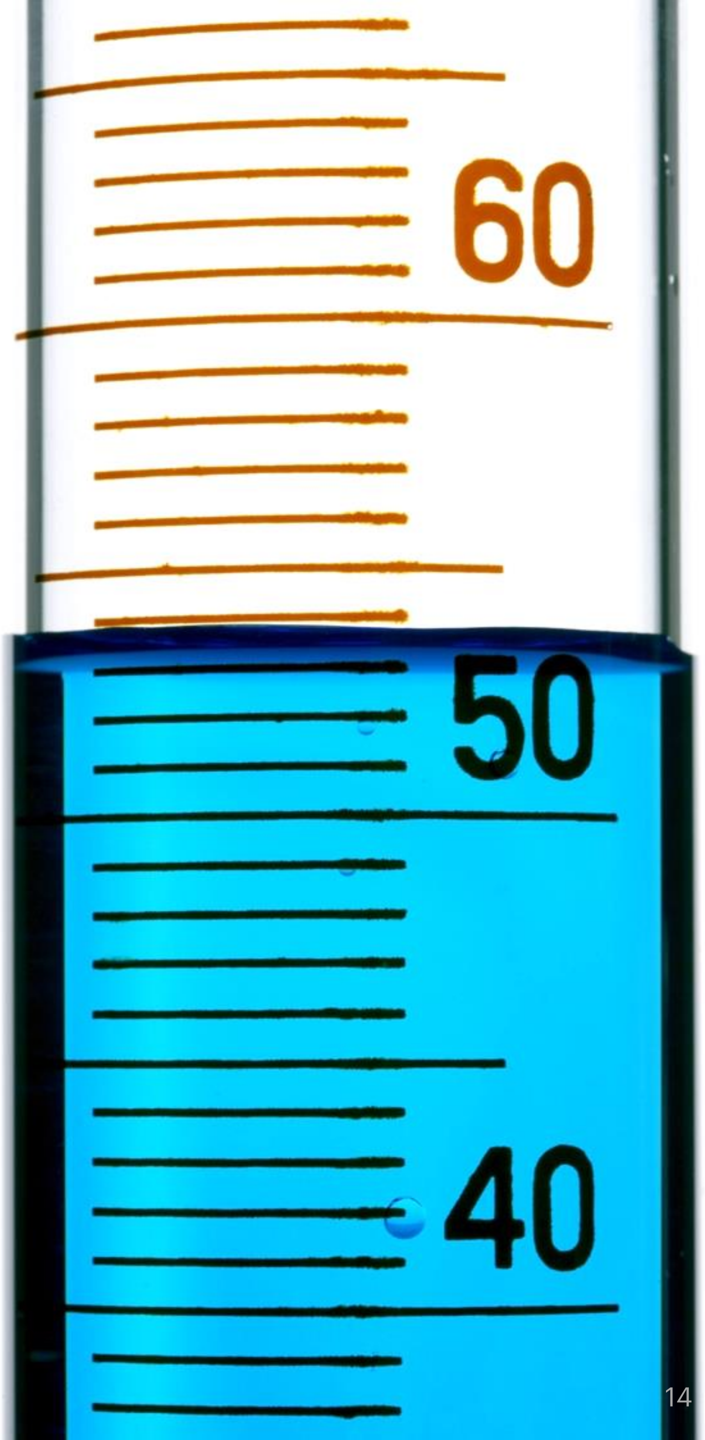


Coverage 89.5%

Overview	
On new code	
Coverage	95.0%
Lines to Cover	136
Uncovered Lines	10
Line Coverage	92.6%
Conditions to Cover	64
Uncovered Conditions	0
Condition Coverage	100%
Overall	
Coverage	89.5%
Lines to Cover	2,152
Uncovered Lines	182
Line Coverage	91.5%
Conditions to Cover	1,036
Uncovered Conditions	154
Condition Coverage	85.1%
Tests	
Unit Tests	279

src/main/java/deors/core/commons/AbstractFileTool.java
src/main/java/deors/core/commons/ConfigurationToolkit.java
src/main/java/deors/core/commons/inifile/INIFileManager.java
src/main/java/deors/core/commons/classloader/ParentLastURLClassLoa
src/main/java/deors/core/commons/scheduler/SchedulerServlet.java
src/main/java/deors/core/commons/threadpool/AbstractMultiThread.ja
src/main/java/deors/core/commons/log/Log4jMemoryAppender.java
src/main/java/deors/core/commons/scheduler/SchedulerTask.java
src/main/java/deors/core/commons/scheduler/Scheduler.java
src/main/java/deors/core/commons/threadpool/MultiThreadPool.java
src/main/java/deors/core/commons/AbstractContext.java
src/main/java/deors/core/commons/AbstractDaemon.java
src/main/java/deors/core/commons/io/ExtensionFileFilter.java
src/main/java/deors/core/commons/io/IOToolkit.java
src/main/java/deors/core/commons/base64/Base64Toolkit.java

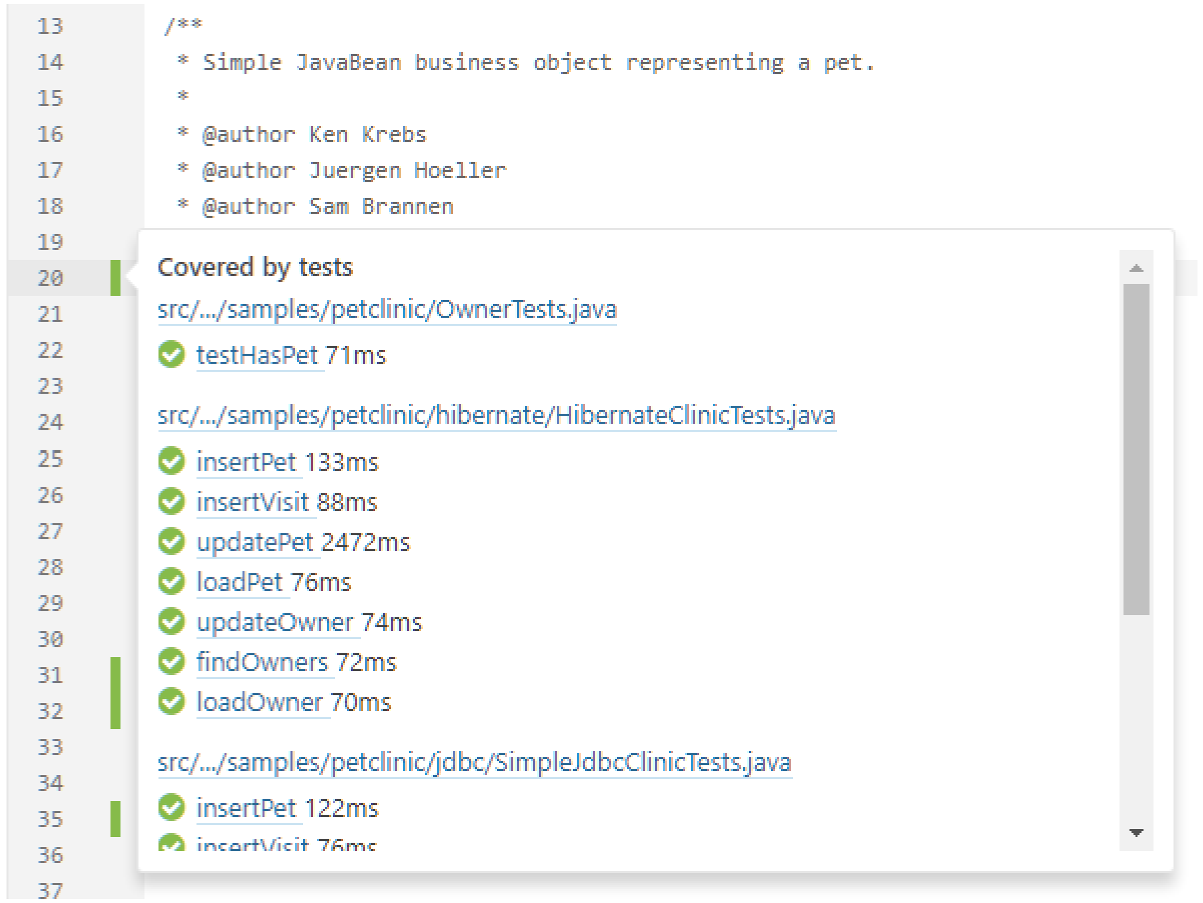
Shows coverage on new code when SonarQube is integrated with version control



Copyright 2018 Accenture. All rights reserved.

CODE COVERAGE TOOLS

Code coverage per test in SonarQube:



```
/**
 * Simple JavaBean business object representing a pet.
 *
 * @author Ken Krebs
 * @author Juergen Hoeller
 * @author Sam Brannen
```

Covered by tests

src/.../samples/petclinic/OwnerTests.java

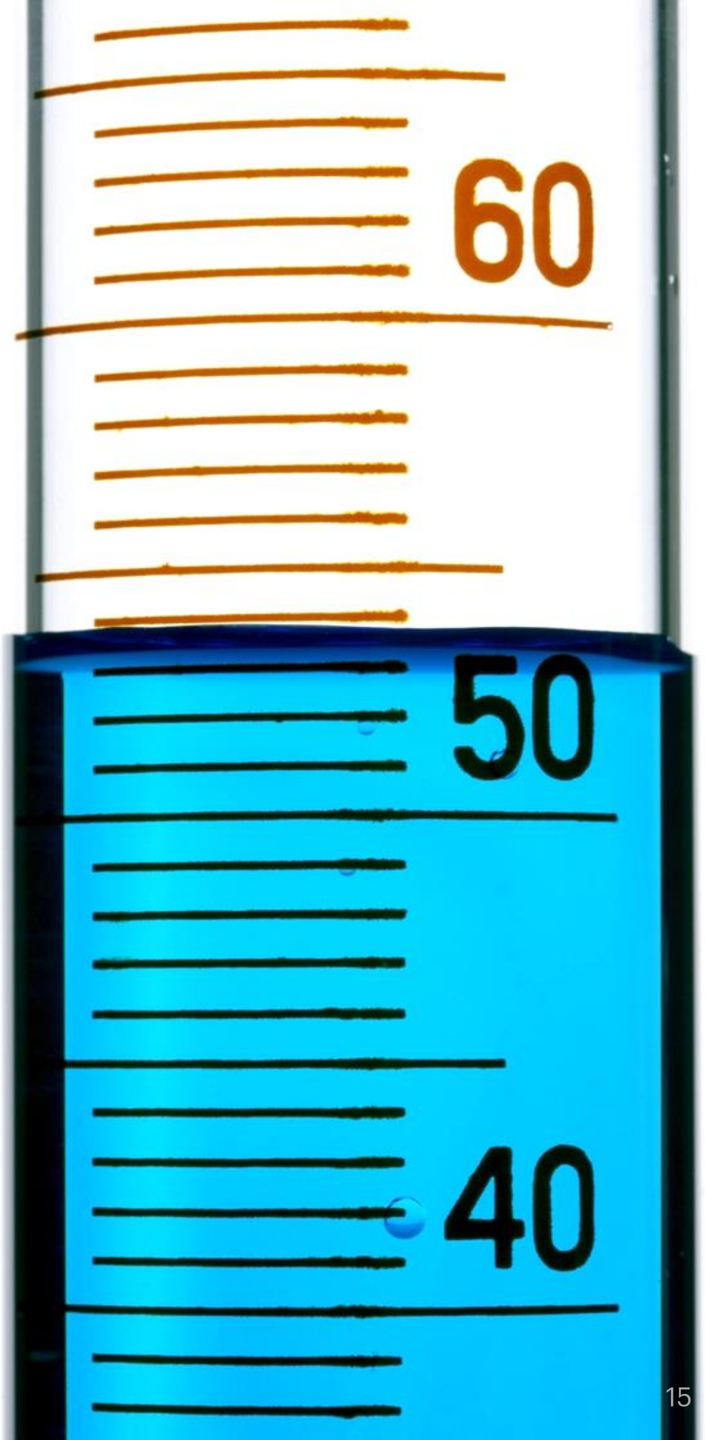
- ✓ testHasPet 71ms

src/.../samples/petclinic/hibernate/HibernateClinicTests.java

- ✓ insertPet 133ms
- ✓ insertVisit 88ms
- ✓ updatePet 2472ms
- ✓ loadPet 76ms
- ✓ updateOwner 74ms
- ✓ findOwners 72ms
- ✓ loadOwner 70ms

src/.../samples/petclinic/jdbc/SimpleJdbcClinicTests.java

- ✓ insertPet 122ms
- ✓ insertVisit 76ms



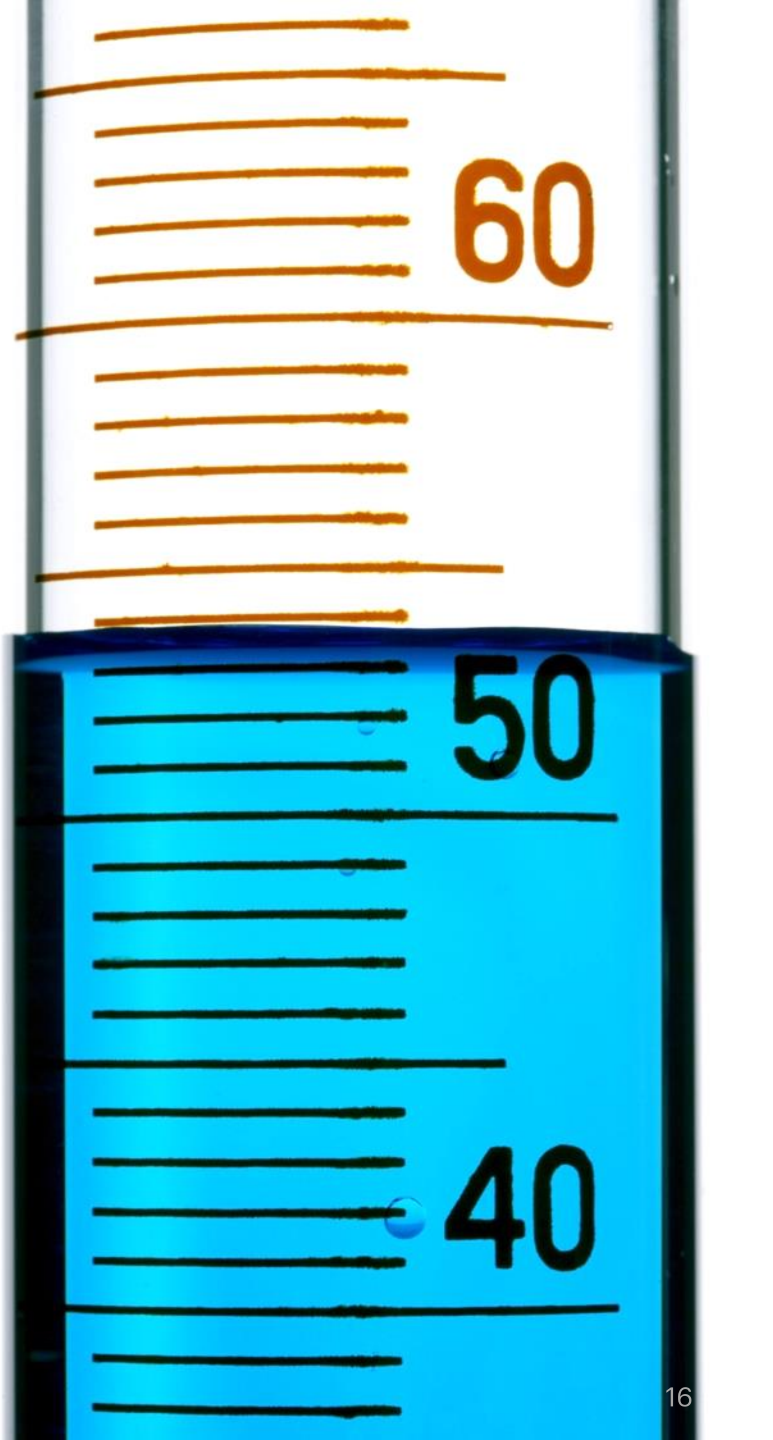
Copyright 2018 Accenture. All rights reserved.

CODE COVERAGE TOOLS

To enable code coverage per test:

1. Add these dependencies to pom.xml

```
<dependency>
    <groupId>org.jacoco</groupId>
    <artifactId>org.jacoco.agent</artifactId>
    <version>0.8.0</version>
    <classifier>runtime</classifier>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.sonarsource.java</groupId>
    <artifactId>sonar-jacoco-listeners</artifactId>
    <version>5.1.0.13090</version>
    <scope>test</scope>
</dependency>
```



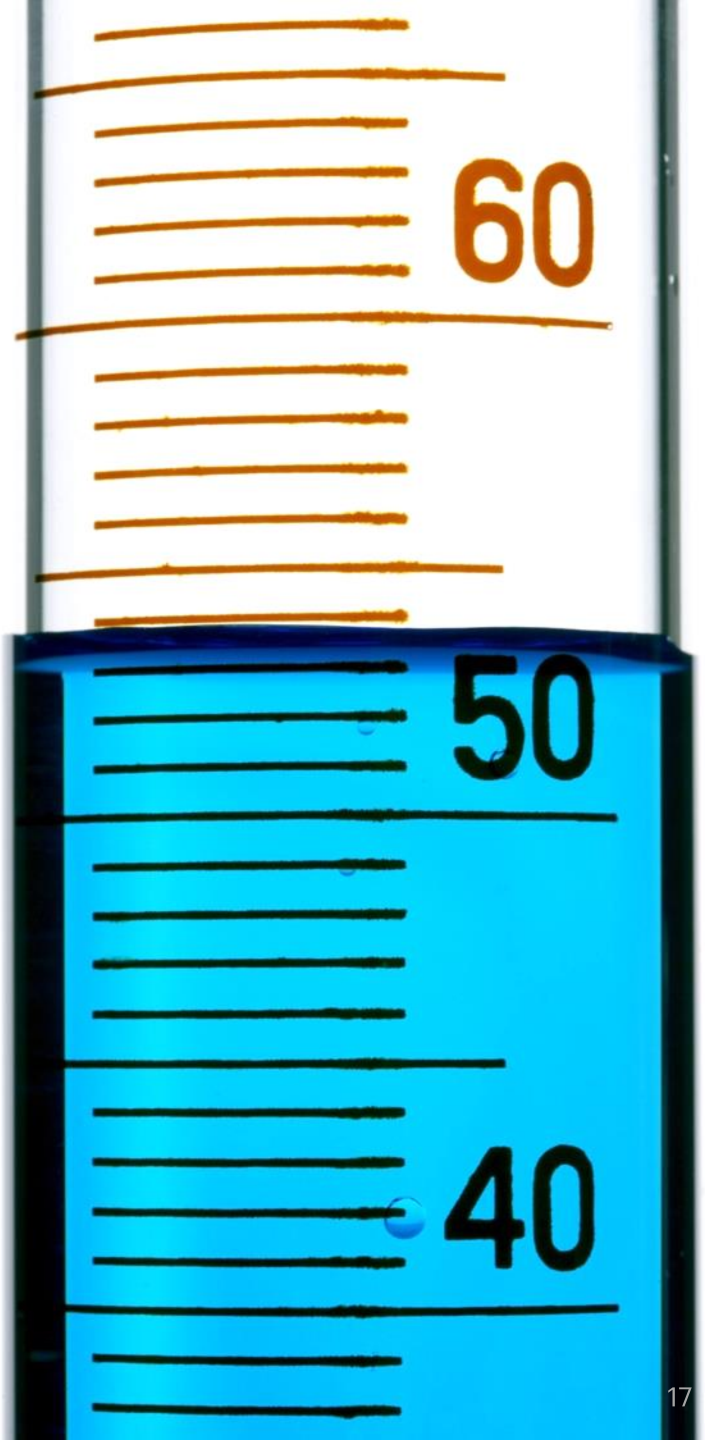
Copyright 2018 Accenture. All rights reserved.

CODE COVERAGE TOOLS

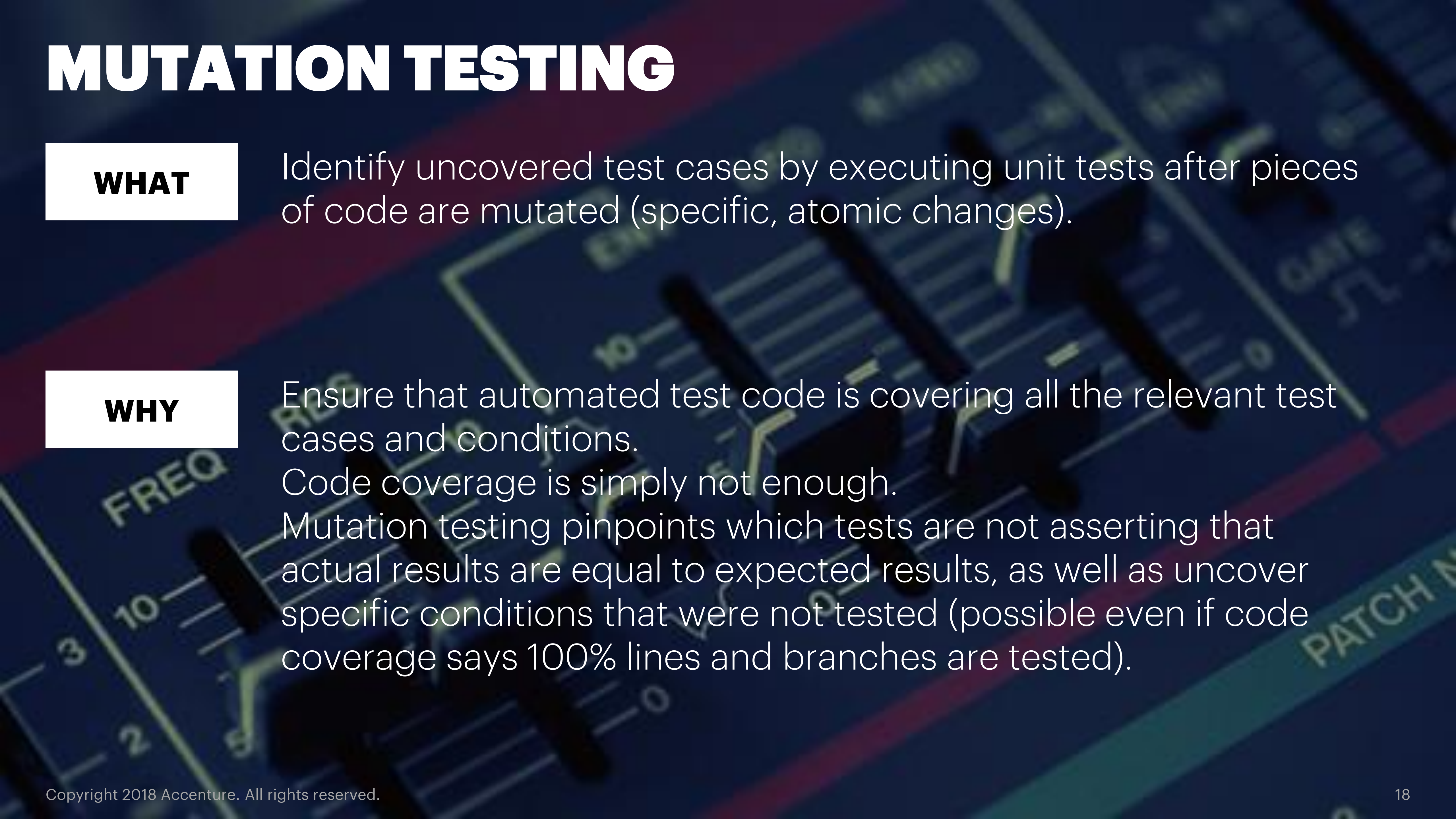
To enable code coverage per test:

2. Enable JaCoCo listener in Surefire

```xml
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-plugin</artifactId>
    <version>2.20.1</version>
    <configuration>
        <properties>
            <property>
                <name>listener</name>
                <value>org.sonar.java.jacoco.JUnitListener</value>
            </property>
        </properties>
    </configuration>
</plugin>
```



Copyright 2018 Accenture. All rights reserved.



MUTATION TESTING

WHAT

Identify uncovered test cases by executing unit tests after pieces of code are mutated (specific, atomic changes).

WHY

Ensure that automated test code is covering all the relevant test cases and conditions.
Code coverage is simply not enough.
Mutation testing pinpoints which tests are not asserting that actual results are equal to expected results, as well as uncover specific conditions that were not tested (possible even if code coverage says 100% lines and branches are tested).

Copyright 2018 Accenture. All rights reserved.

MUTATION TESTING TOOLS

Java	pitest.org
Groovy	
Scala	
JS TS	Stryker

MUTATION TESTING TOOLS

pitest.org

Mutation testing tools introduce controlled changes in application code, one at a time

Code base	Code mutation
if (a >= 0)	if (a < 0)
if (b == 1)	if (a == -1)
someObject.someMethod("hi")	someObject.someMethod(null)
someObject.someMethod(whatever)	Method call is removed

Re-execute those tests executing the modified logic
If tests do not fail, then the test is wrong

- It is not asserting thoroughly enough
- It is not asserting anything!

Copyright 2018 Accenture. All rights reserved.

MUTATION TESTING TOOLS

Key facts for the really impatient:

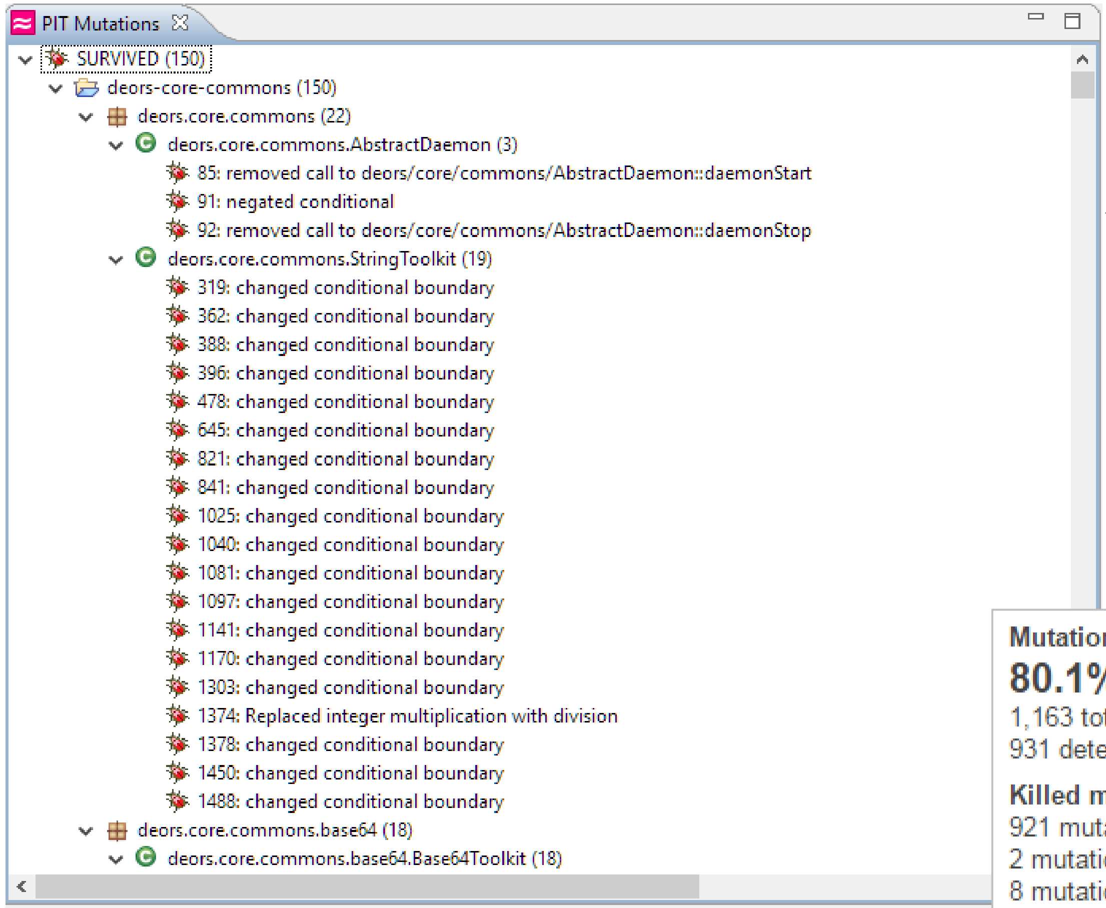
It does not require changes in application code
It does not require changes in test code
It requires zero or little configuration
It mutates on bytecodes, so it is as efficient as possible
It re-executes only the relevant tests after a change

Yet… it takes time!

Run Pitest in the background in your IDE, or in CI builds publishing results to SonarQube!

Copyright 2018 Accenture. All rights reserved.

MUTATION TESTING TOOLS

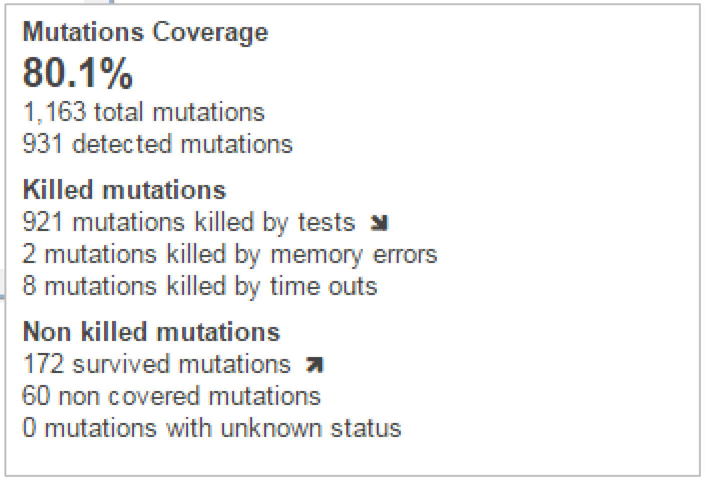


PIT Mutations

```
SURVIVED (150)
  deors-core-commons (150)
    deors.core.commons (22)
      deors.core.commons.AbstractDaemon (3)
        85: removed call to deors/core/commons/AbstractDaemon::daemonStart
        91: negated conditional
        92: removed call to deors/core/commons/AbstractDaemon::daemonStop
      deors.core.commons.StringToolkit (19)
        319: changed conditional boundary
        362: changed conditional boundary
        388: changed conditional boundary
        396: changed conditional boundary
        478: changed conditional boundary
        645: changed conditional boundary
        821: changed conditional boundary
        841: changed conditional boundary
        1025: changed conditional boundary
        1040: changed conditional boundary
        1081: changed conditional boundary
        1097: changed conditional boundary
        1141: changed conditional boundary
        1170: changed conditional boundary
        1303: changed conditional boundary
        1374: Replaced integer multiplication with division
        1378: changed conditional boundary
        1450: changed conditional boundary
        1488: changed conditional boundary
    deors.core.commons.base64 (18)
      deors.core.commons.base64.Base64Toolkit (18)
```

Results in Eclipse

Results in SonarQube



Mutations Coverage
80.1%
1,163 total mutations
931 detected mutations

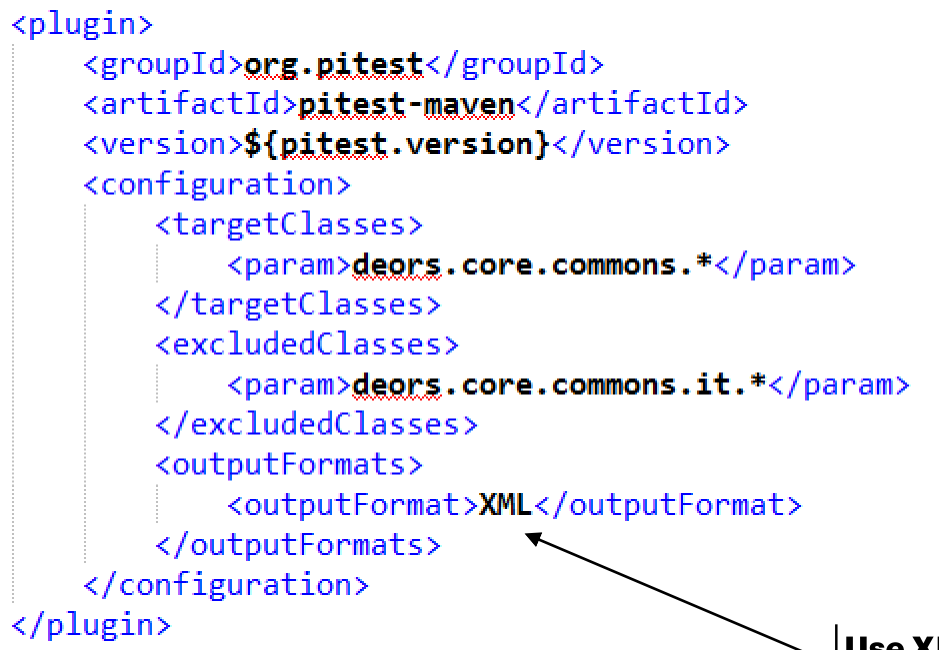
Killed mutations
921 mutations killed by tests
2 mutations killed by memory errors
8 mutations killed by time outs

Non killed mutations
172 survived mutations
60 non covered mutations
0 mutations with unknown status

Copyright 2018 Accenture. All rights reserved.

MUTATION TESTING TOOLS

Configure filters wisely, Pitest can take very long hours to execute, e.g. with integration tests



```xml
<plugin>
    <groupId>org.pitest</groupId>
    <artifactId>pitest-maven</artifactId>
    <version>${pitest.version}</version>
    <configuration>
        <targetClasses>
            <param>deors.core.commons.*</param>
        </targetClasses>
        <excludedClasses>
            <param>deors.core.commons.it.*</param>
        </excludedClasses>
        <outputFormats>
            <outputFormat>XML</outputFormat>
        </outputFormats>
    </configuration>
</plugin>
```

Use XML output to pull data into SonarQube

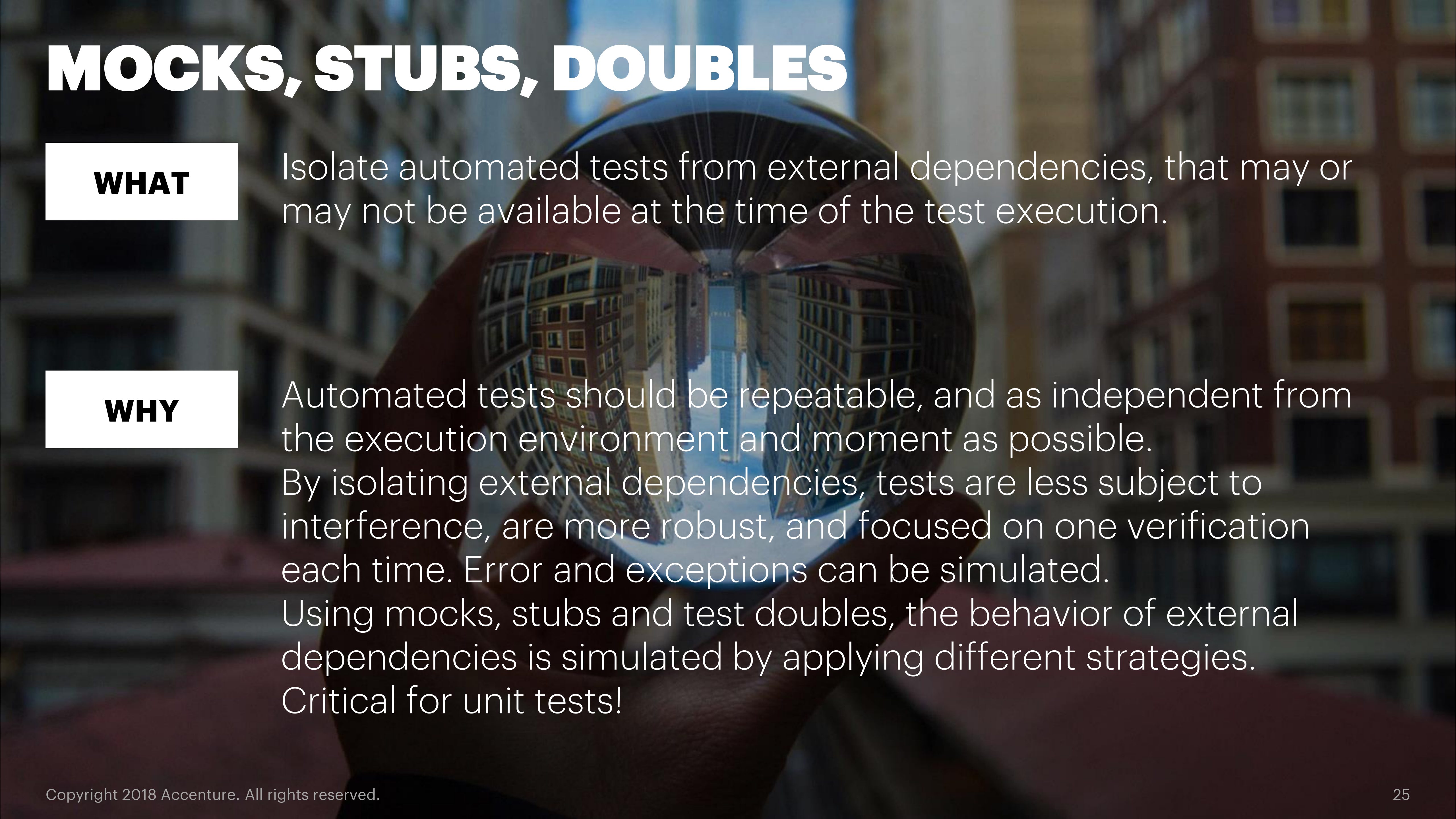
Copyright 2018 Accenture. All rights reserved.

MUTATION TESTING TOOLS

Example mutators:

- Conditionals Boundary Mutator
- Negate Conditionals Mutator
- Remove Conditionals Mutator
- Math Mutator
- Increments Mutator
- Invert Negatives Mutator
- Inline Constant Mutator
- Return Values Mutator
- Void Method Calls Mutator
- Non Void Method Calls Mutator
- Constructor Calls Mutator

Copyright 2018 Accenture. All rights reserved.



MOCKS, STUBS, DOUBLES

WHAT

Isolate automated tests from external dependencies, that may or may not be available at the time of the test execution.

WHY

Automated tests should be repeatable, and as independent from the execution environment and moment as possible.
By isolating external dependencies, tests are less subject to interference, are more robust, and focused on one verification each time. Error and exceptions can be simulated.
Using mocks, stubs and test doubles, the behavior of external dependencies is simulated by applying different strategies.
Critical for unit tests!

Copyright 2018 Accenture. All rights reserved.

MOCKING FRAMEWORKS

Java

EASYMOCK WM mockito

PowerMock jMock JMockit

Groovy

Spock

EASYMOCK mockito

jMock JMockit

Scala

ScalaMock

EASYMOCK mockito

jMock JMockit

JS TS

SINON.JS

MOCKING FRAMEWORKS

EasyMock provides common mocking patterns

PowerMock is capable of instrumenting code and make testable code that isn't:

- static blocks
- constructors
- object instantiation
- private members

JMockit provides all capabilities above combined, with a more modern and expressive API

Copyright 2018 Accenture. All rights reserved.

MOCKING FRAMEWORKS

Constructor that opens an LDAP connection

```
public class DirectoryManager {

    public DirectoryManager(String directoryHost, int directoryPort)
        throws DirectoryException {

        super();

        if (directoryHost == null || directoryHost.length() == 0 || directoryPort <= 0) {
            throw new IllegalArgumentException("ERR_OPEN_CONN_ARG");
        }

        try {
            connection = new LDAPConnection();
            connection.connect(directoryHost, directoryPort);
        } catch (LDAPException ldape) {
            throw new DirectoryException("ERR_OPEN_CONN", ldape);
        }

        connected = true;
    }
…
}
```

Copyright 2018 Accenture. All rights reserved.

MOCKING FRAMEWORKS

Making it testable with EasyMock + PowerMock

```
@RunWith(PowerMockRunner.class)
@PrepareForTest(DirectoryManager.class)
public class DirectoryManagerPowerMockTestCase {

    @Test(expected = DirectoryException.class)
    public void testConstructorError() throws Exception {

        LDAPConnection lc = PowerMock.createMock(LDAPConnection.class);
        PowerMock.expectNew(LDAPConnection.class).andReturn(lc);
        lc.connect("localhost", 2000);
        EasyMock.expectLastCall().andThrow(new LDAPException("error", 1, "error"));

        PowerMock.replay(lc, LDAPConnection.class);

        new DirectoryManager("localhost", 2000);
    }
…
}
```

Copyright 2018 Accenture. All rights reserved.

MOCKING FRAMEWORKS

Making it testable with JMockit

```
@RunWith(JMockit.class)
public class DirectoryManagerJMockitTestCase {

    @Mocked(stubOutClassInitialization = true)
    LDAPConnection connection = new LDAPConnection();

    @Test(expected = DirectoryException.class)
    public void testConstructorError() throws Exception {

        new Expectations() {{
            connection.connect("localhost", 2000);
            result = new LDAPException("error", 1, "error");
        }};

    new DirectoryManager("localhost", 2000);
}
```

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING

WHAT

Analyze code, both statically and dynamically, to identify potential security issues: vulnerabilities, defensive programming patterns, etc.

WHY

As applications grow in complexity, and as more and more services are directly exposed to end consumers over the Internet, and as we speed up the release processes thanks to DevOps, it is adamant to have automated security tests along the life-cycle.
Prevent impersonation, personal and sensible information leaks (passwords, social security numbers, credit card data), business confidential information, secret reports, etc.
Scans look at both source code and external dependencies!

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

Java	sonarqube	FindBugs™	Find Security Bugs	SpotBugs	
		OWASP ZAP		OWASP Dependency Check	
Groovy		OWASP ZAP		OWASP Dependency Check	
Scala		OWASP ZAP		OWASP Dependency Check	
JS TS		OWASP ZAP		OWASP Dependency Check	

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

OWASP **ZAP**

Dynamic profiler – Two modes:

#1 Passive Scan

Works as an HTTP proxy

Analyzes HTTP requests and responses (for example, during test execution, ideally automated in a CI/CD pipeline)

Looks for known vulnerabilities like:

- ★ SQL injection
- ★ Cross site request forgery (CSRF)
- ★ Cross site scripting (XSS)
- ★ Cookie handling

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

OWASP **ZAP**

Dynamic profiler – Two modes:

#2 Active Scan

Launch coordinated attacks on the target application

It should be executed only in applications you are authorized to

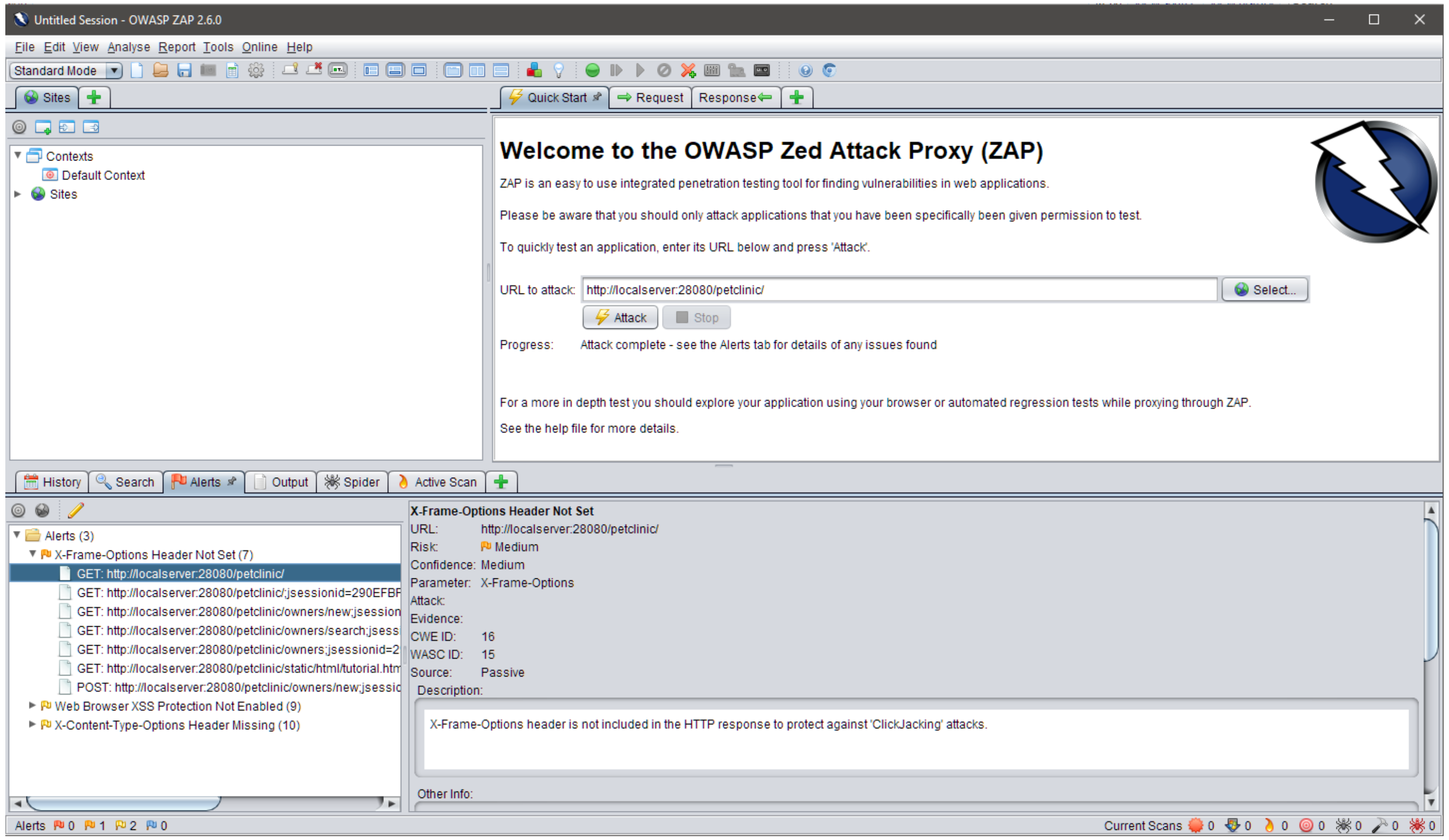
Never in production, it can break things, and lead to data loss

It may take a long, long time to complete a full scan, even in a simple application

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

OWASP **ZAP**



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://localserver:28080/petclinic/

Attack Stop

Progress: Attack complete - see the Alerts tab for details of any issues found

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

See the help file for more details.

Alerts (3)
- X-Frame-Options Header Not Set (7)
  - GET: http://localserver:28080/petclinic/
  - GET: http://localserver:28080/petclinic/;jsessionid=290EFBF
  - GET: http://localserver:28080/petclinic/owners/new;jsession
  - GET: http://localserver:28080/petclinic/owners/search;jsess
  - GET: http://localserver:28080/petclinic/owners;jsessionid=2
  - GET: http://localserver:28080/petclinic/static/html/tutorial.htm
  - POST: http://localserver:28080/petclinic/owners/new;jsessic
- Web Browser XSS Protection Not Enabled (9)
- X-Content-Type-Options Header Missing (10)

X-Frame-Options Header Not Set
URL: http://localserver:28080/petclinic/
Risk: Medium
Confidence: Medium
Parameter: X-Frame-Options
Attack:
Evidence:
CWE ID: 16
WASC ID: 15
Source: Passive
Description:

X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

Other Info:

Alerts 0 1 2 0 Current Scans 0 0 0 0 0 0 0

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

OWASP **Dependency Check**

Scans dependencies for a given project/module, looking for known vulnerabilities in those dependencies (version-wise)

Uses NIST *National Vulnerability Database* (NVD)

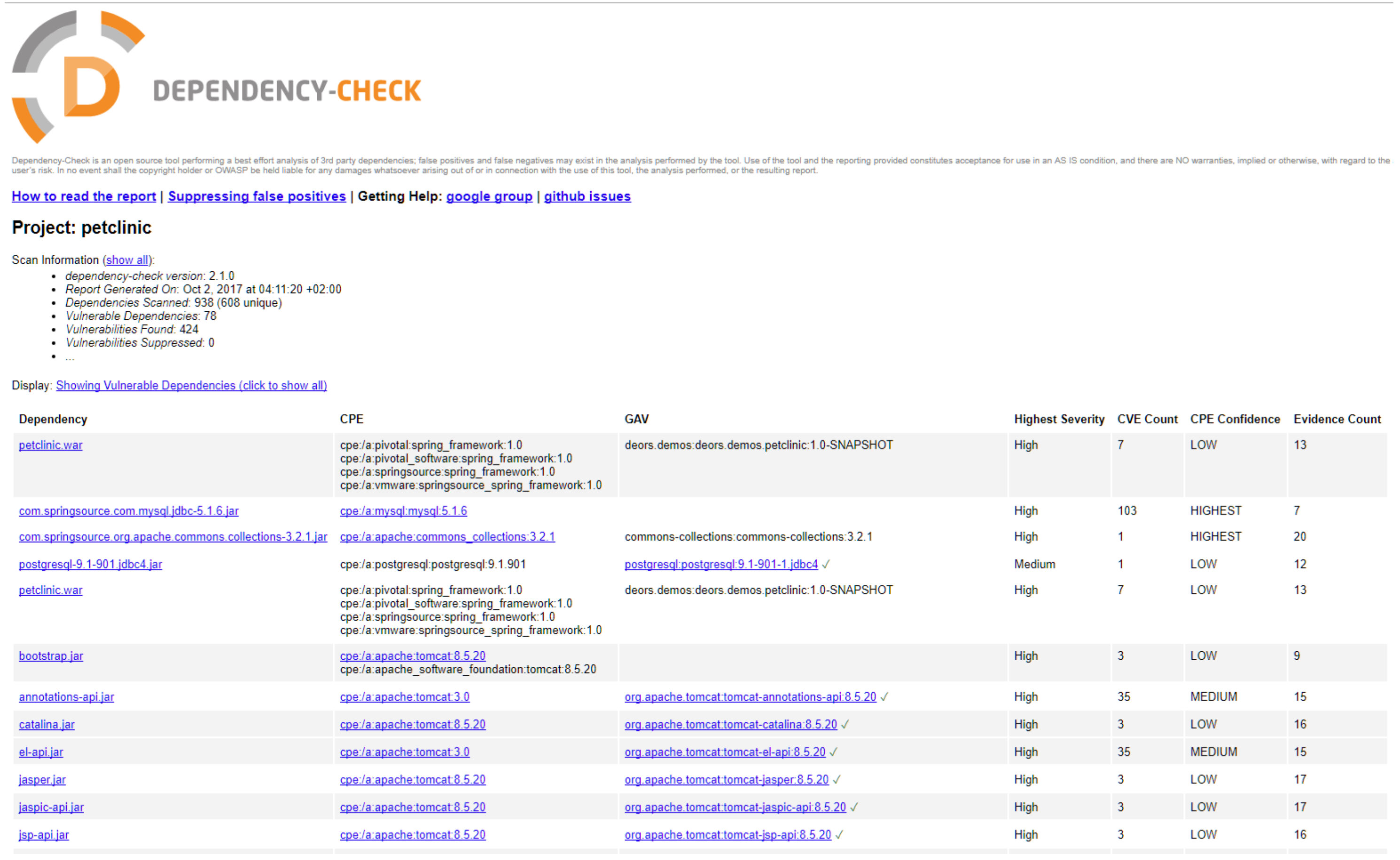
Can be run from command-line, Ant, Maven, Gradle, sbt or Jenkins

Can be integrated with SonarQube

Copyright 2018 Accenture. All rights reserved.

SECURITY TESTING TOOLS

OWASP Dependency Check



DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the user's risk. In no event shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: google group | github issues

Project: petclinic

Scan Information (show all):

- *dependency-check version*: 2.1.0
- *Report Generated On*: Oct 2, 2017 at 04:11:20 +02:00
- *Dependencies Scanned*: 938 (608 unique)
- *Vulnerable Dependencies*: 78
- *Vulnerabilities Found*: 424
- *Vulnerabilities Suppressed*: 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
petclinic.war	cpe:/a:pivotal:spring_framework:1.0 cpe:/a:pivotal_software:spring_framework:1.0 cpe:/a:springsource:spring_framework:1.0 cpe:/a:vmware:springsource_spring_framework:1.0	deors.demos:deors.demos.petclinic:1.0-SNAPSHOT	High	7	LOW	13
com.springsource.com.mysql.jdbc-5.1.6.jar	cpe:/a:mysql:mysql:5.1.6		High	103	HIGHEST	7
com.springsource.org.apache.commons.collections-3.2.1.jar	cpe:/a:apache:commons_collections:3.2.1	commons-collections:commons-collections:3.2.1	High	1	HIGHEST	20
postgresql-9.1-901.jdbc4.jar	cpe:/a:postgresql:postgresql:9.1.901	postgresql:postgresql:9.1-901-1.jdbc4 ✓	Medium	1	LOW	12
petclinic.war	cpe:/a:pivotal:spring_framework:1.0 cpe:/a:pivotal_software:spring_framework:1.0 cpe:/a:springsource:spring_framework:1.0 cpe:/a:vmware:springsource_spring_framework:1.0	deors.demos:deors.demos.petclinic:1.0-SNAPSHOT	High	7	LOW	13
bootstrap.jar	cpe:/a:apache:tomcat:8.5.20 cpe:/a:apache_software_foundation:tomcat:8.5.20		High	3	LOW	9
annotations-api.jar	cpe:/a:apache:tomcat:3.0	org.apache.tomcat:tomcat-annotations-api:8.5.20 ✓	High	35	MEDIUM	15
catalina.jar	cpe:/a:apache:tomcat:8.5.20	org.apache.tomcat:tomcat-catalina:8.5.20 ✓	High	3	LOW	16
el-api.jar	cpe:/a:apache:tomcat:3.0	org.apache.tomcat:tomcat-el-api:8.5.20 ✓	High	35	MEDIUM	15
jasper.jar	cpe:/a:apache:tomcat:8.5.20	org.apache.tomcat:tomcat-jasper:8.5.20 ✓	High	3	LOW	17
jaspic-api.jar	cpe:/a:apache:tomcat:8.5.20	org.apache.tomcat:tomcat-jaspic-api:8.5.20 ✓	High	3	LOW	17
jsp-api.jar	cpe:/a:apache:tomcat:8.5.20	org.apache.tomcat:tomcat-jsp-api:8.5.20 ✓	High	3	LOW	16

SUMMARY

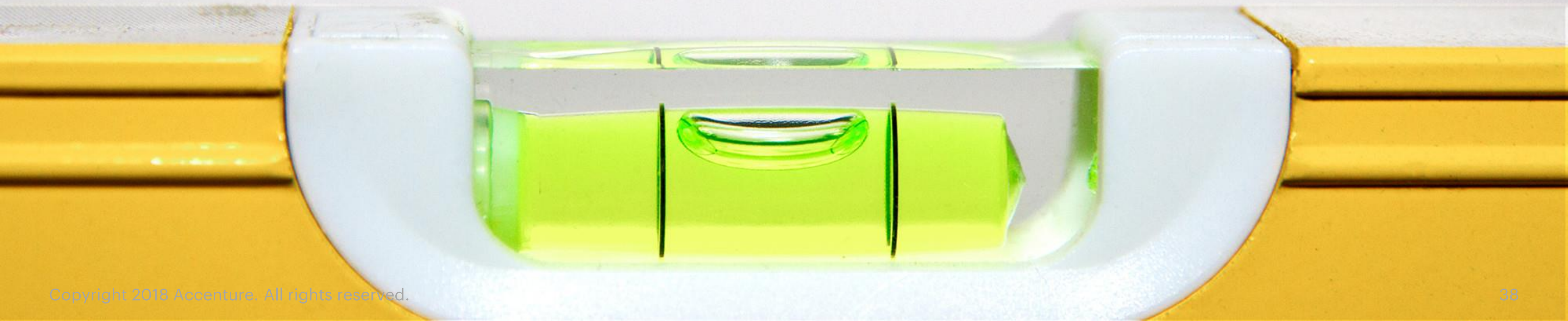
PROFILE YOUR CODE
Pick a static code profiler to automate review of coding standards and common best practices

MEASURE COVERAGE
Understand which parts of your code are not being tested by mixing code coverage and mutation testing

MOCKS ARE GOOD
They help to make tests repeatable and independent from the environment, and make testable, code that isn't

SECURITY FIRST
Put security first by combining defensive programming patterns with checks from static and dynamic profilers

PIPELINES
Run all these tools continuously in CI/CD pipelines



Copyright 2018 Accenture. All rights reserved.

REFERENCES

SonarQube – https://www.sonarqube.org

ESLint – https://eslint.org

EclEmma & JaCoCo – http://www.eclemma.org

Pitest – http://pitest.org

JMockit – http://jmockit.org

FindSecBugs – https://find-sec-bugs.github.io

OWASP ZAP – https://www.owasp.org/index.php/ZAP

OWASP Dependency Check – https://www.owasp.org/index.php/OWASP_Dependency_Check

Copyright 2018 Accenture. All rights reserved.