

# Quo pertentas, OSS?

How Open Source can benefit from well-crafted  
Tests

**Björn Kimminich**

Web: <http://kimminich.de>

Twitter: @bkimminich

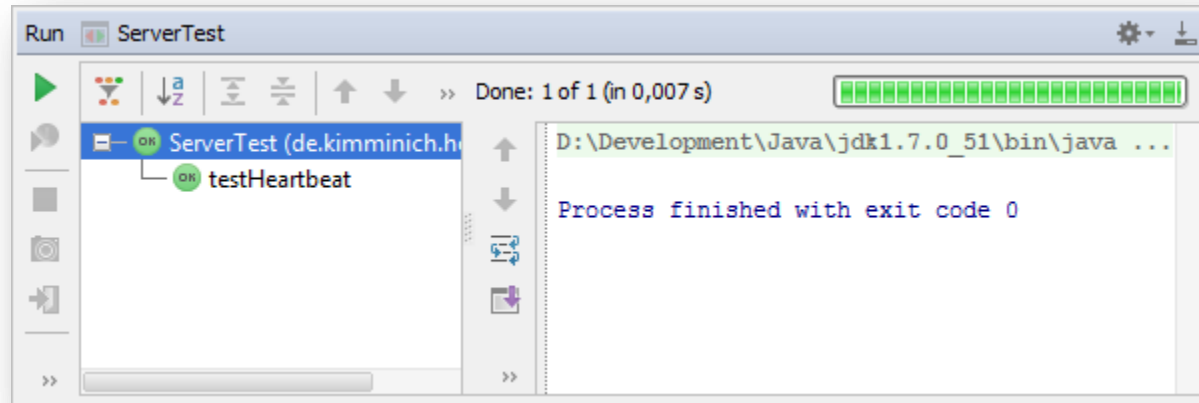
# Let's start with some code...

```
public class Server {  
  
    private String memory;  
  
    public Server(String username, String password) {  
        this.memory = this.hashCode() + ";user=" + username + "&password=" + password + ";" + this.getClass().getName();  
    }  
  
    public String heartbeat(String payload, int length) {  
        memory = "pl=" + payload + memory;  
        return memory.substring(3, 3+length);  
    }  
  
}
```

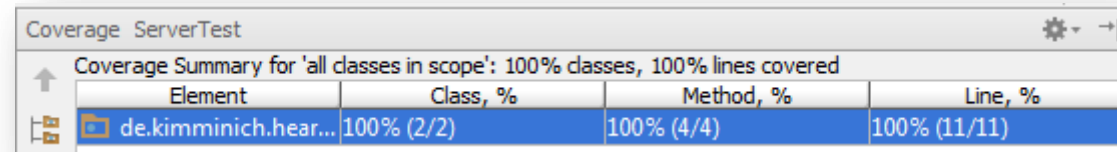
# ...and a corresponding unit test!

```
@Test
public void testHeartbeat() throws Exception {
    Server server = new Server("bjoern.kimminich", "s3cret123!");
    String payload = "1100101";
    String reply = server.heartbeat(payload, payload.length());
    assertEquals(payload, reply);
}
```

# It passes with flying colors...

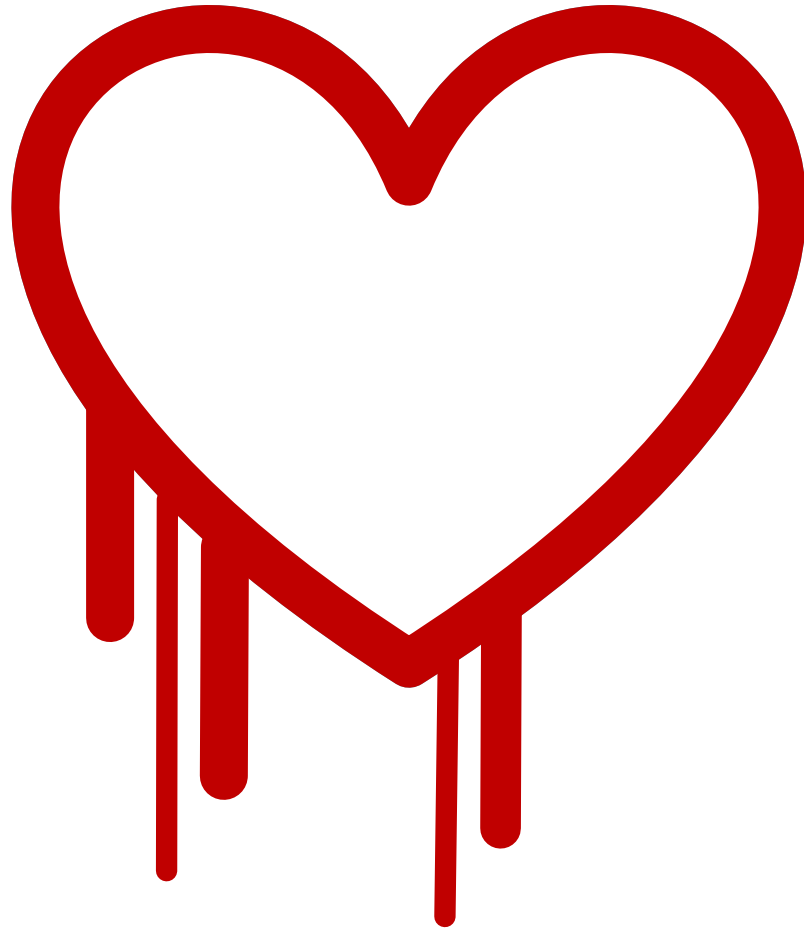


... and achieves 100% code coverage!



Coverage Summary for 'all classes in scope': 100% classes, 100% lines covered			
Element	Class, %	Method, %	Line, %
de.kimminich.hear...	100% (2/2)	100% (4/4)	100% (11/11)

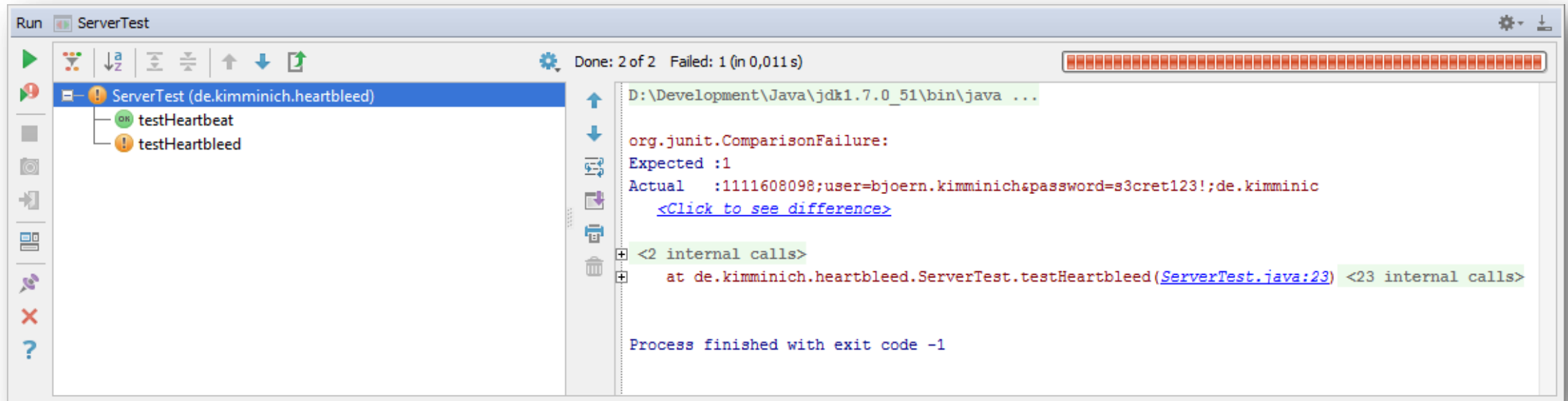
Nothing could possibly go wrong!



# How about adding another test?

```
@Test
public void testHeartbleed() throws Exception {
    Server server = new Server("bjoern.kimminich", "s3cret123!");
    String payload = "1";
    String reply = server.heartbeat(payload, 64);
    assertEquals(payload, reply);
}
```

# Oops!





# Finding Bugs in Open Source Software



# Code Reviews



## Pair Programming

Infeasible with remote development

Occasionally during Hackathons



## Peer Review

Developers review each other

Hard to organize properly



## Committer Review

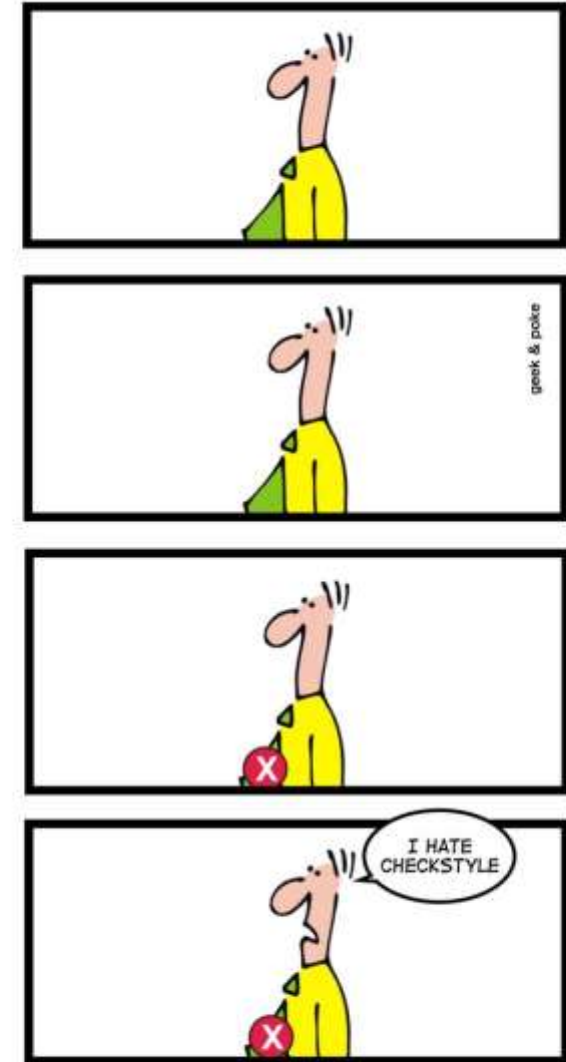
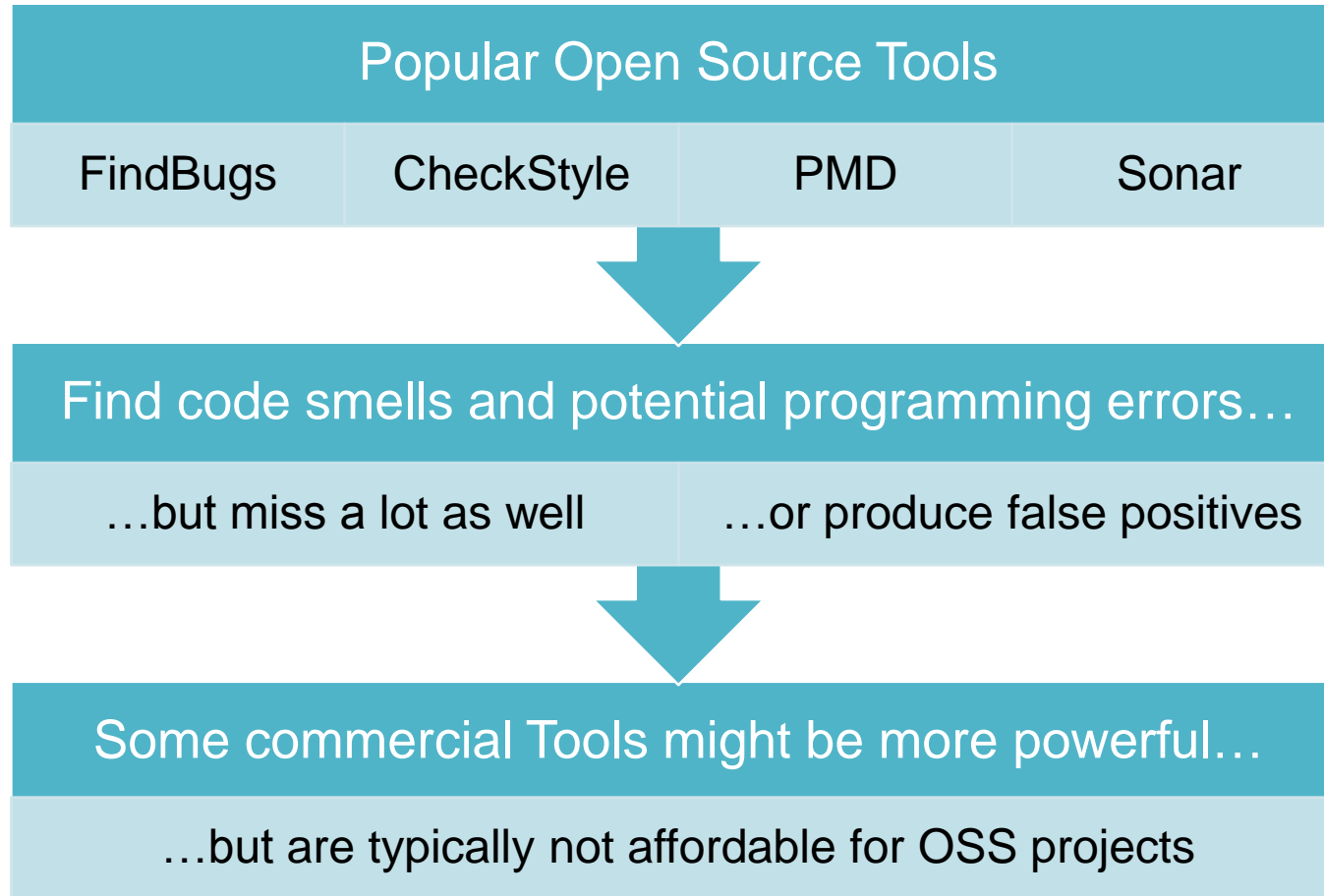
Not everyone has commit rights

Senior developers review contributions before merge into master

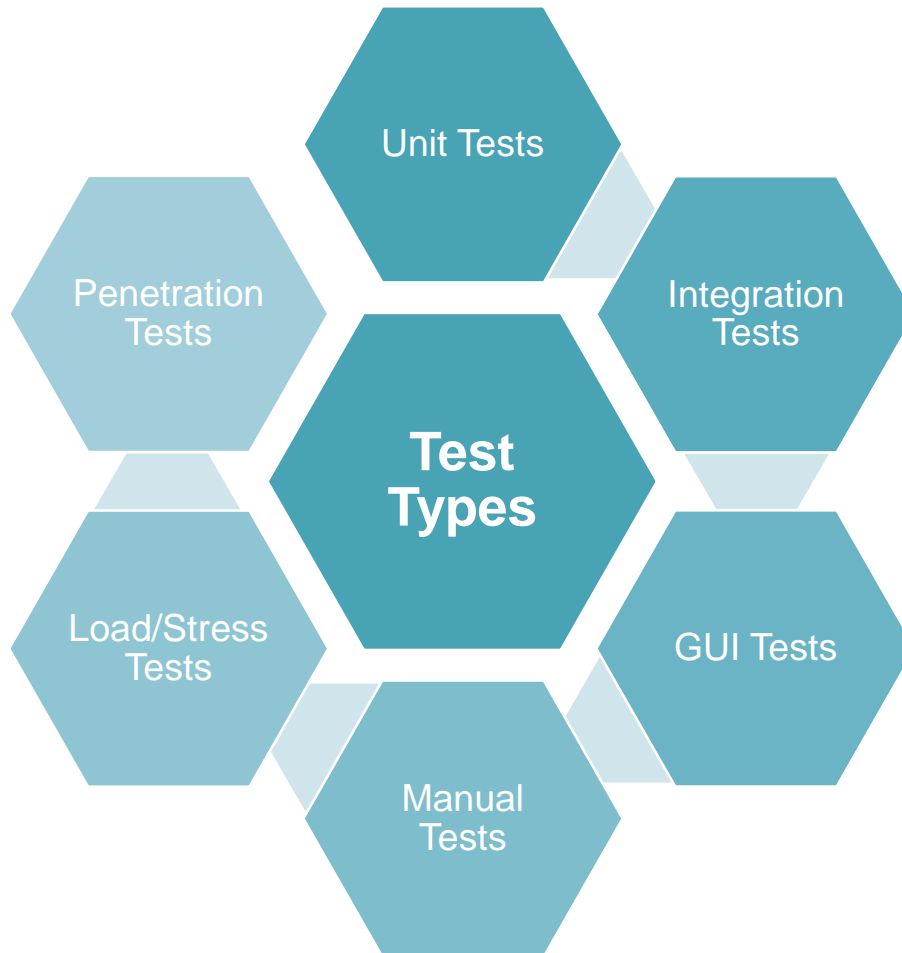
## HOW TO MAKE A GOOD CODE REVIEW



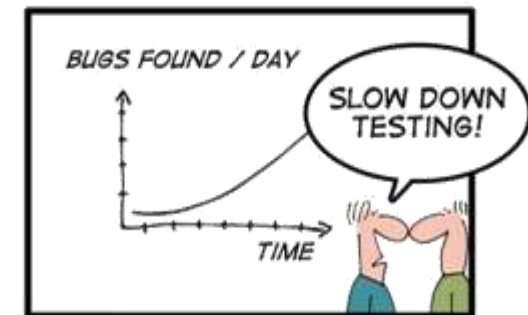
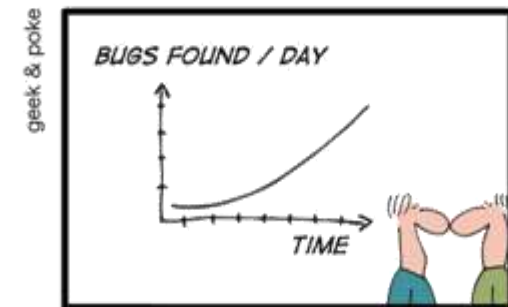
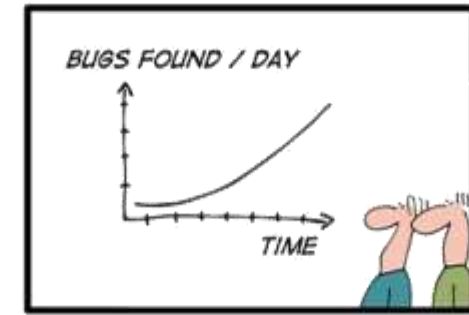
# Static Code Analysis



# Testing



PROJECT MANAGEMENT MADE EASY

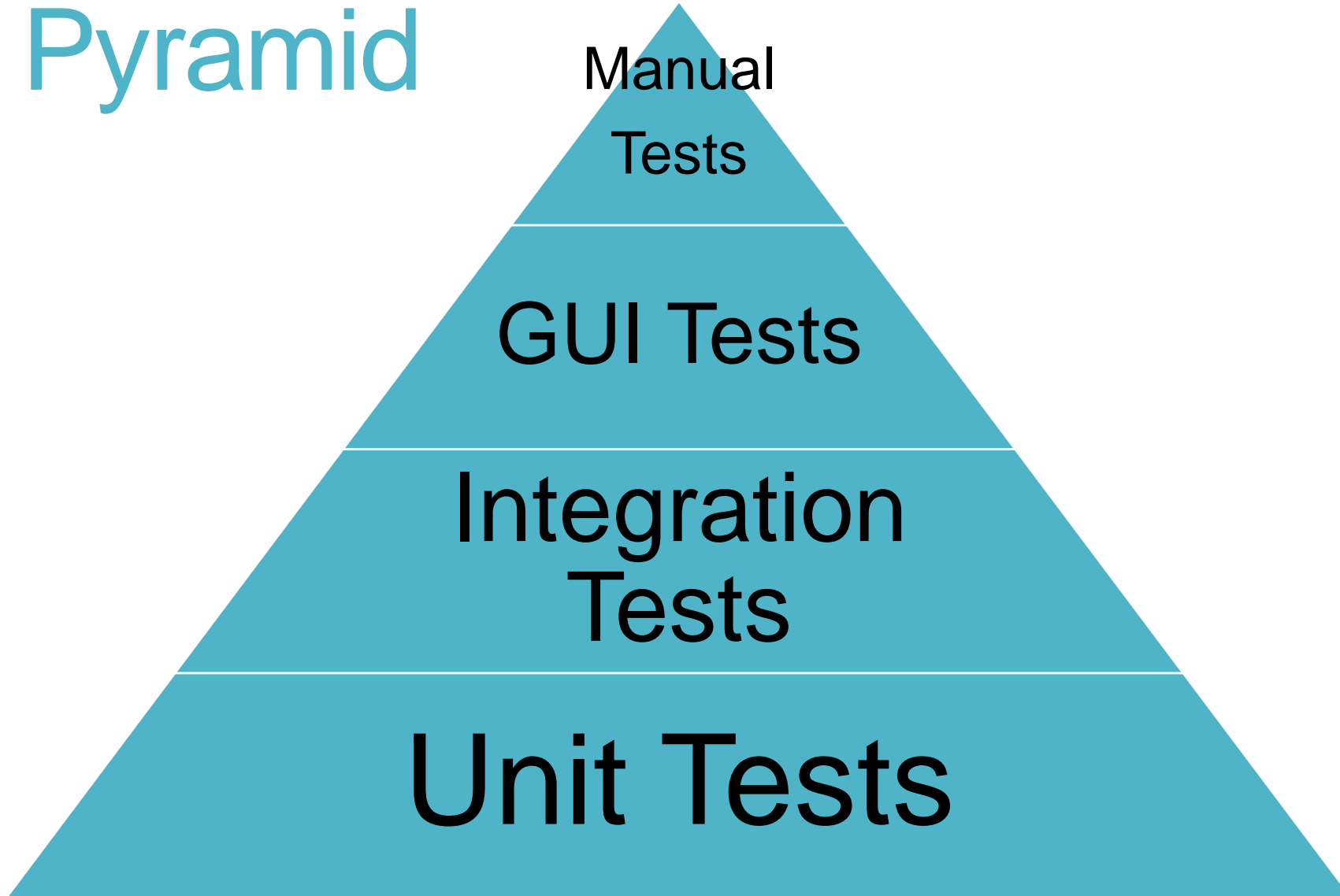


TEST MANAGEMENT

# Best vs. Bad Practices for Testing

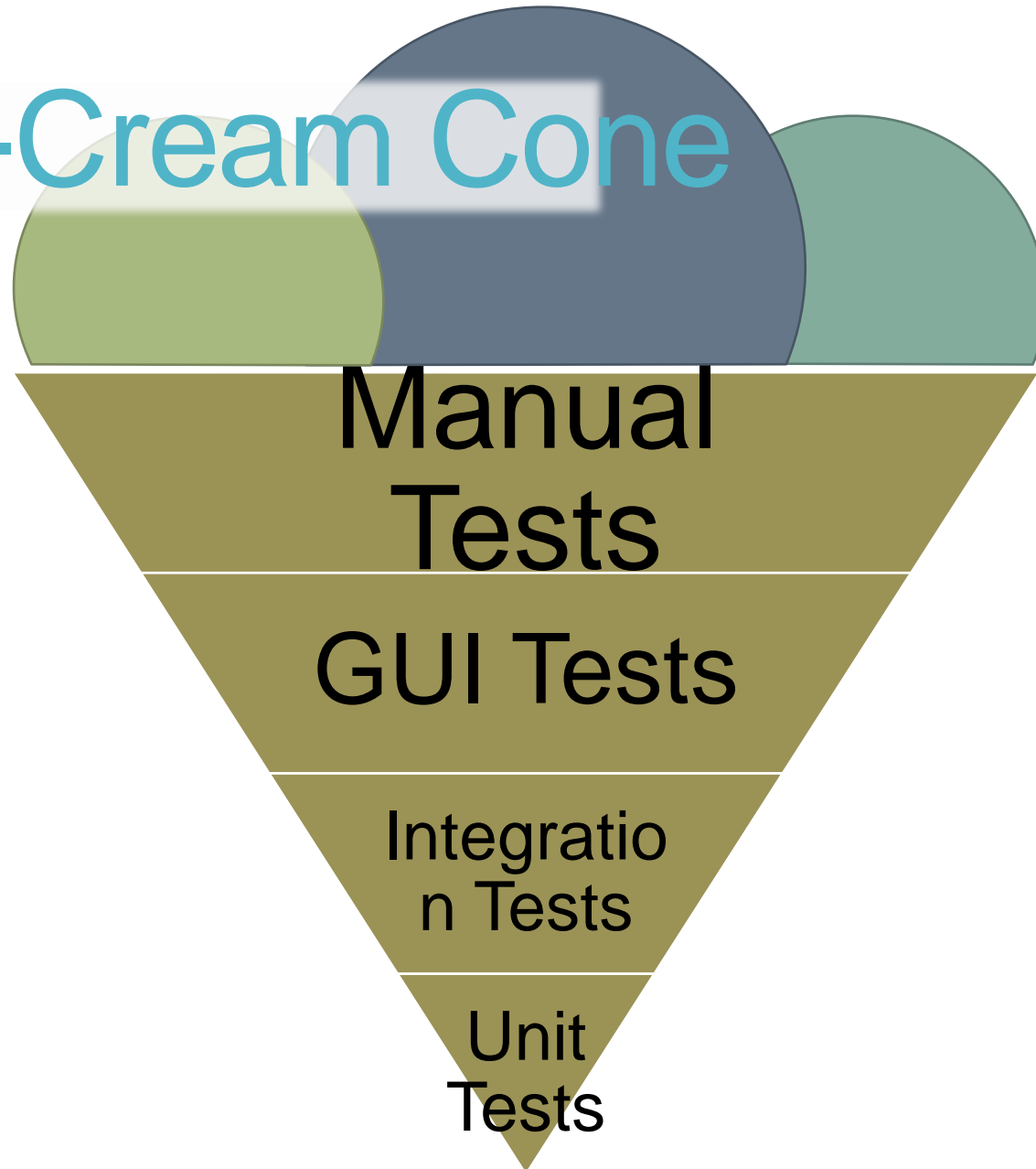


# Test Pyramid



PRACTICE BEST

# Test Ice-Cream Cone



PRACTICE BAD



# Happy Path Testing

BAD  
PRACTICE



Photo: [Tortured Mind Photography](#)



# Testing Border & Exceptional Cases

PRACTICE

```
@Test(expected = IllegalArgumentException.class)
public void shouldThrowExceptionOnMissingBaseUrl() { URLResolver.resolveUrl(null, "notNull"); }

@Test(expected = IllegalArgumentException.class)
public void shouldThrowExceptionOnMissingRelativeUrl() { URLResolver.resolveUrl("notNull", null); }

@Test
public void shouldAppendRelativeUrlToBaseUrlHost() {
    assertThat(URLResolver.resolveUrl("http://www.abc.de", "/xy/z"), is("http://www.abc.de/xy/z"));
}

@Test
public void shouldInsertSlashBetweenBaseUrlAndRelativeUrlIfMissing() {
    assertThat(URLResolver.resolveUrl("http://www.abc.de", "xyz"), is("http://www.abc.de/xyz"));
}

@Test
public void shouldReplaceLastPartOfUrlPathFromBaseUrlWithRelativeUrl() {
    assertThat(URLResolver.resolveUrl("http://www.abc.de/w/x", "y/z"), is("http://www.abc.de/w/y/z"));
}
```

# No Assertions

PRACTICE  
BAD

```
public void testAllMethods() throws Exception {
    // create a user to test Anonymous
    String accountName = ESAPI.randomizer().getRandomString(8, EncoderConstants.CHAR_ALPHANUMERICS);
    Authenticator instance = ESAPI.authenticator();
    String password = instance.generateStrongPassword();

    // Probably could skip the assignment here, but maybe someone had
    // future plans to use this. So will just suppress warning for now.
    /unused/
    User user = instance.createUser(accountName, password, password);

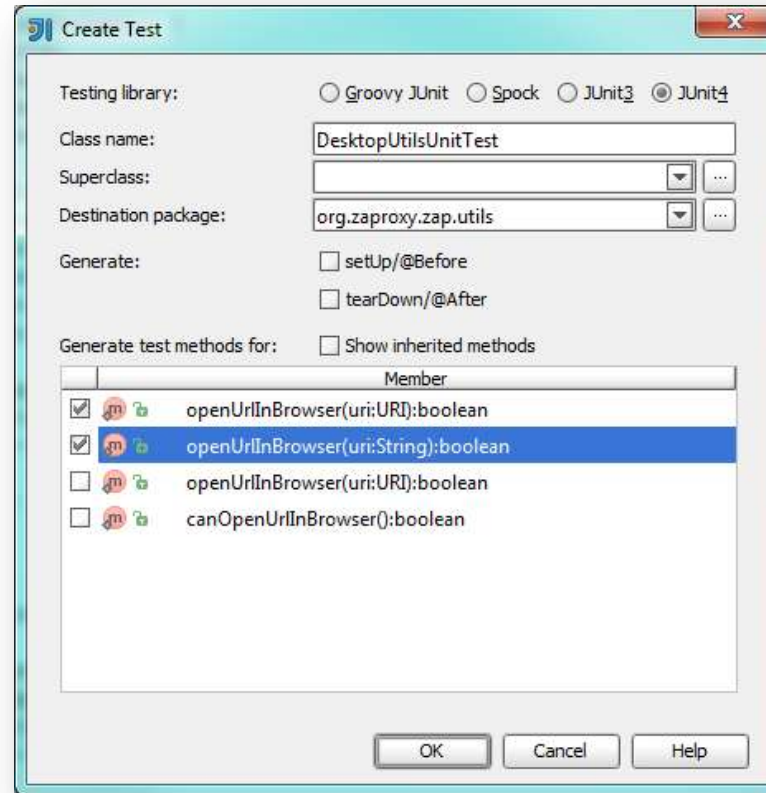
    // test the rest of the Anonymous user
    try { User.ANONYMOUS.addRole(null); } catch( RuntimeException e ) {}
    try { User.ANONYMOUS.addRoles(null); } catch( RuntimeException e ) {}
    try { User.ANONYMOUS.changePassword(null, null, null); } catch( RuntimeException e ) {}

    /* [...] */

    try { User.ANONYMOUS.getAccountName(); } catch( RuntimeException e ) {}
    try { User.ANONYMOUS.getAccountName(); } catch( RuntimeException e ) {}
}
```

# API Tests

PRACTICE  
BAD



# Scenario Tests with BDD

PRACTICE

```
@Test
public void compressedResponseBodyShouldBeDeflatedIntoApiResponse() throws Exception {
    given(responseHeader.getHeader(HttpHeader.CONTENT_ENCODING)).willReturn(HttpHeader.GZIP);
    given(responseBody.getBytes()).willReturn(gzip(new byte[] {97, 98, 99}));

    ApiResponseSet response = ApiResponseConversionUtils.httpMessageToSet(0, message);

    assertThat(response.getValues(), hasEntry("responseBody", (Object) "abc"));
}

@Test
public void brokenCompressedResponseBodyShouldBeStoredAsStringRepresentationInApiResponse() {
    given(responseHeader.getHeader(HttpHeader.CONTENT_ENCODING)).willReturn(HttpHeader.GZIP);
    given(responseBody.getBytes()).willReturn(new byte[] {0, 0, 0});

    ApiResponseSet response = ApiResponseConversionUtils.httpMessageToSet(0, message);

    assertThat(response.getValues(), hasEntry("responseBody", (Object) responseBody.toString()));
}
```

Benefits of well-crafted Tests for OSS



# Maintainability++

---

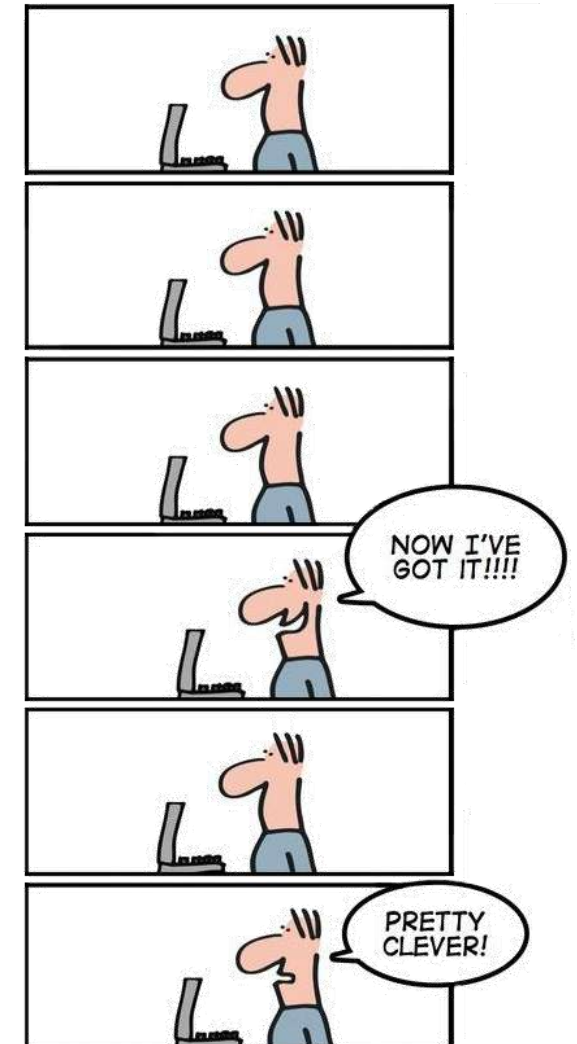
A suite of automated regression tests helps finding defects resulting from code changes

---

New contributors do not have to fear touching old code...

---

...neither do long-time committers after a longer vacation!



*ISN'T IT A GREAT  
FEELING WHEN YOU  
FINALLY GET YOUR OWN  
CODE, YOU'VE WRITTEN  
MONTHS BEFORE?*

# Documentation++

---

External and Javadoc documentation tends to rot quickly and becomes obsolete or even misleading

---

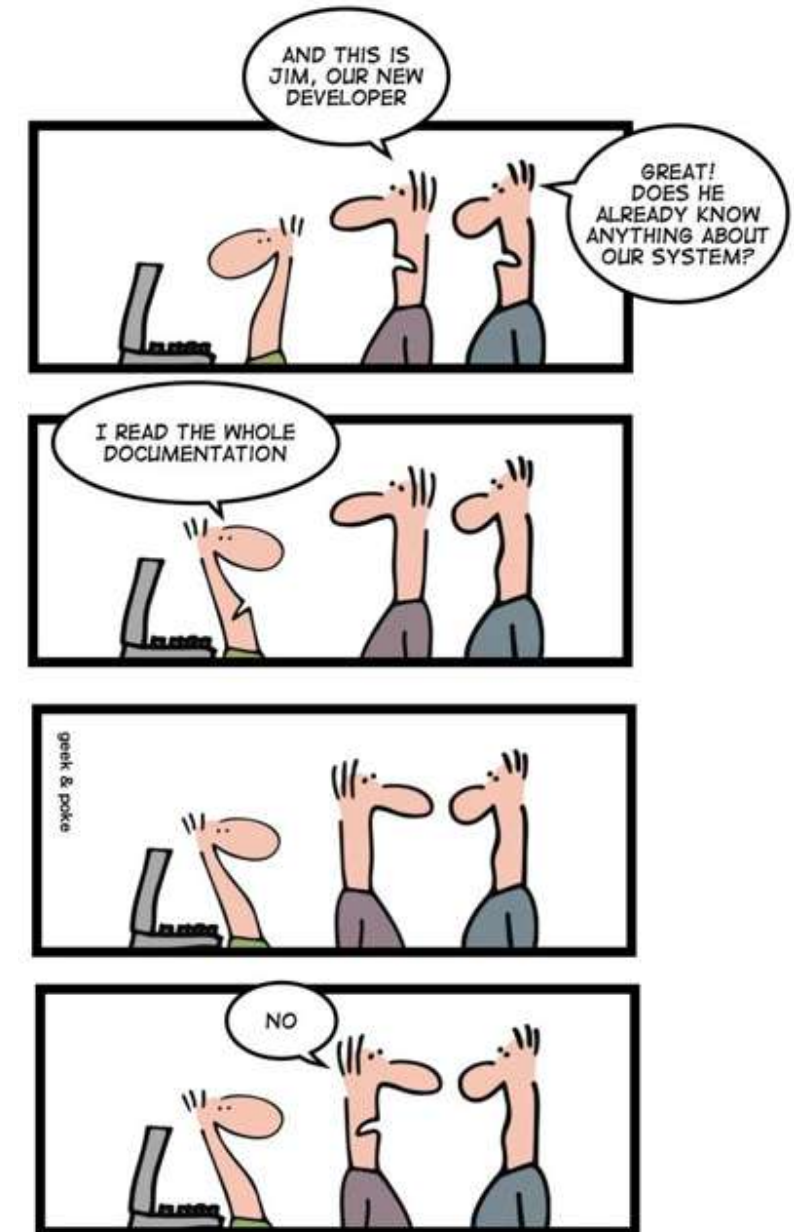
Tests that get outdated tend to break, so they have to be fixed resulting in updated documentation

---

Well-written tests document the intended behavior of a class or component

---

Even if the production code is hard to understand, a good test can help to fill this gap





# Specification++

---

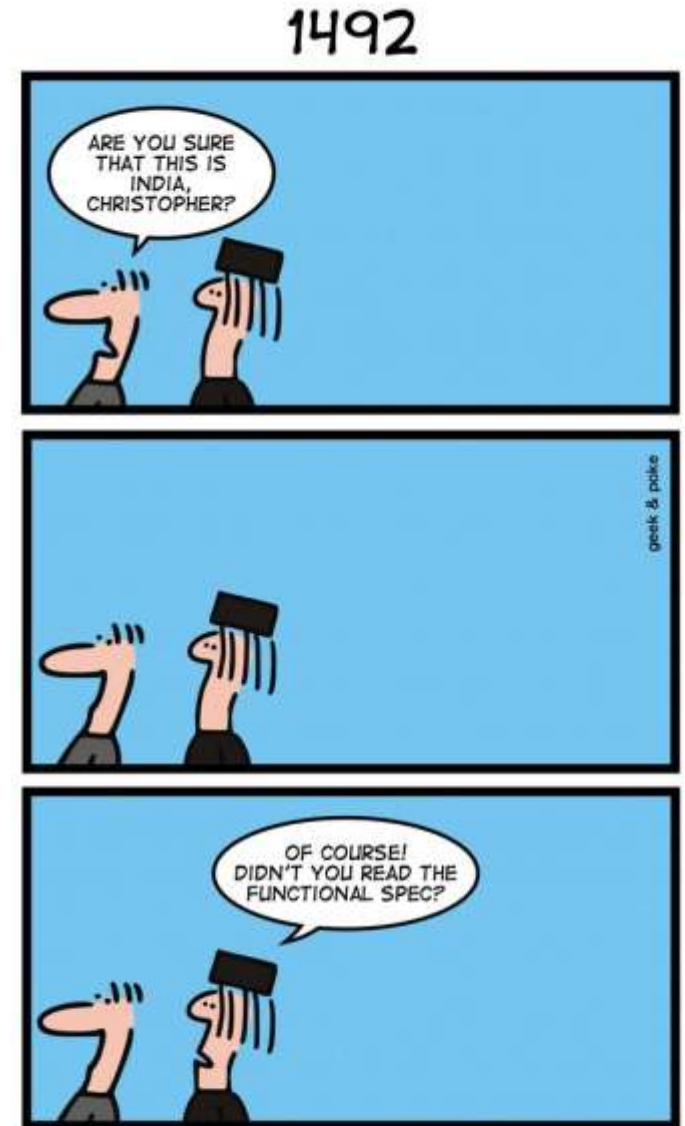
Writing tests before the production code is even better than just documenting existing code

---

Consequent TDD / BDD will let the Tests become the actual specification of the program's intended behavior

---

Failing tests indicate that the specification is not met yet (or any more)

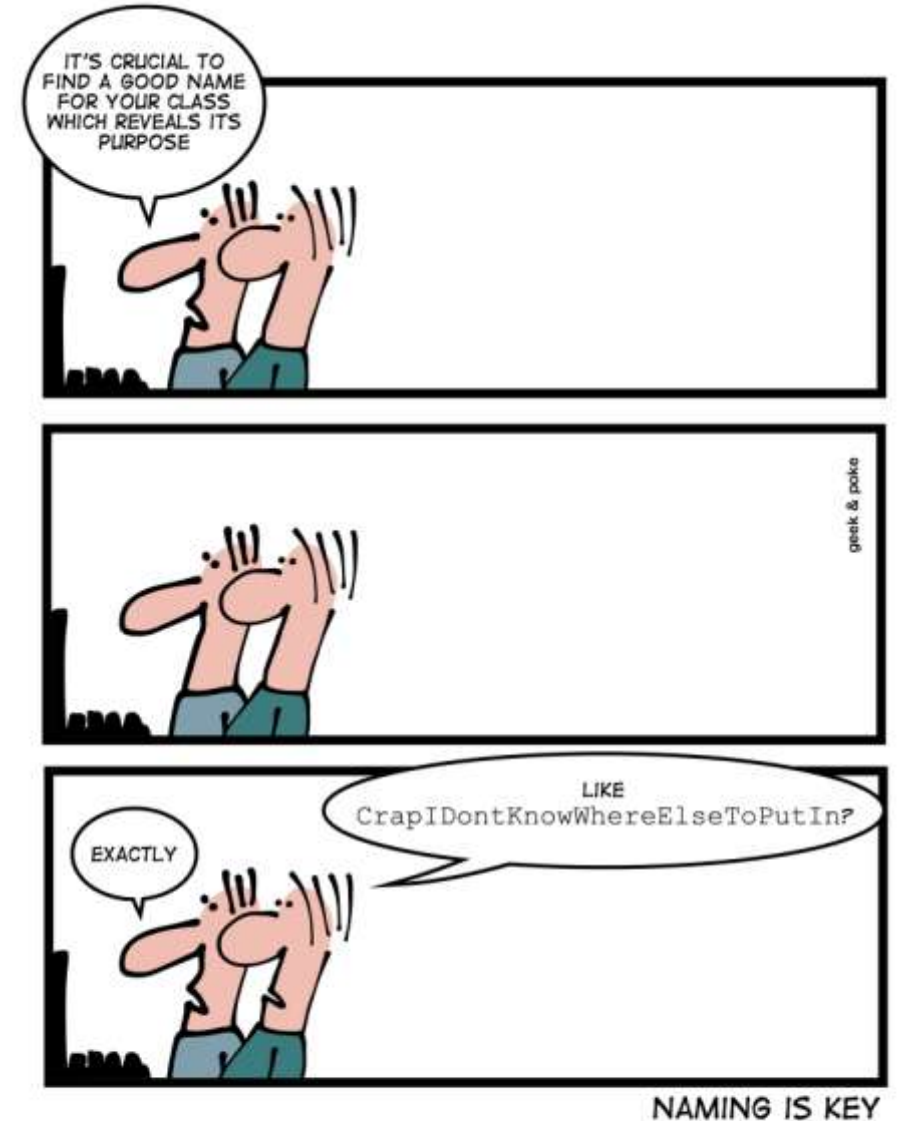




# Contribution++

Well maintained, documented and tested projects are safer and more fun to contribute to

Nobody likes working on an untested piece of unreadable code (especially in their free time)



# Truck Factor++

---

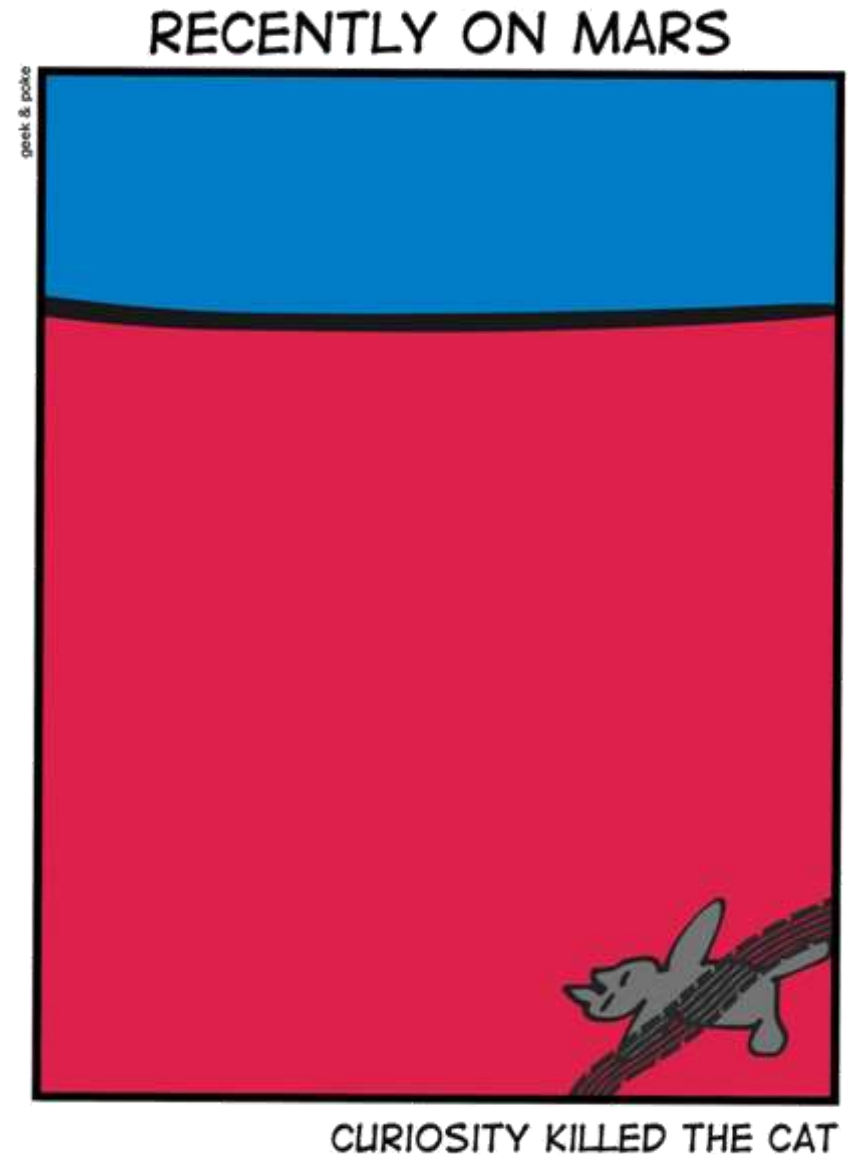
How many project contributors could be fatally hit by a truck before the project perishes?

---

The lower the number, the more volatile the project as it relies on individual experts

---

The number can be increased by spreading knowledge and lowering entry barriers



# Introducing Unit Tests to OWASP ZAP



# OWASP Zed Attack Proxy (ZAP)



---

Easy-to-use  
integrated  
penetration-  
testing tool

Locates vulnerabilities in web  
applications

---

Helps building secure apps

---

OWASP Flagship Project

---

Programmed in Java with *javafx.swing* UI

---

# How to contribute to ZAP?



---

**Develop core features** <https://code.google.com/p/zaproxy/>

---

**Develop addons** <https://code.google.com/p/zap-extensions/>


































---

**Help with translation** <https://crowdin.net/project/owasp-zap>

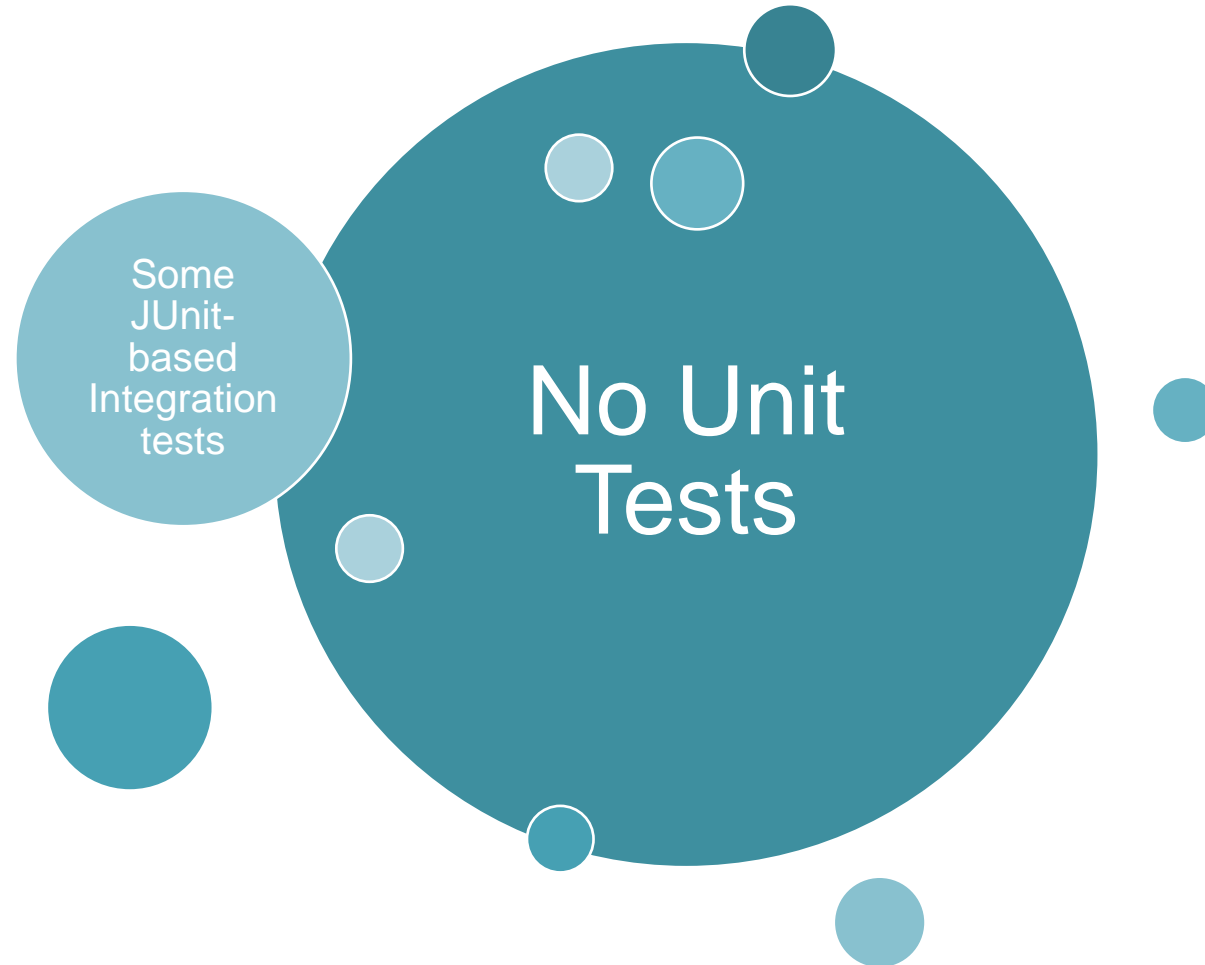
---

**Promote ZAP**  
<https://code.google.com/p/zaproxy/wiki/ZapEvangelists>

# ZAP Truck Factor $\leq 2$

Name	Kudos	12 Month Commits	All Time Commits	5 Year Trend	Primary Language	First Commit	Last Commit
 <b>Simon Bennetts</b> (ZAP Project Lead)		693	1607		Java	over 3 years ago	6 days ago
 <b>THC...@gmail.com</b>		438	719		Java	over 2 years ago	6 days ago
 <b>colm.p.o...@gmail.com</b>		29	241		Java	almost 2 years ago	8 days ago
 <b>Cosmin Stefan Dobrin</b>		127	186		Java	almost 2 years ago	25 days ago
 <b>yha...@gmail.com</b>		176	176		Java	10 months ago	26 days ago
 <b>Björn Kimminich</b>		40	97		Java	over 1 year ago	30 days ago
 <b>maw...@ymail.com</b>		0	76		Java	about 3 years ago	over 1 year ago
 <b>70poi...@gmail.com</b>		74	74		Java	4 months ago	8 days ago
 <b>desousa.vitor</b>		0	67		Java	about 2 years ago	about 2 years ago
 <b>leandro...@talsoft.c...</b>		0	59		Java	over 1 year ago	over 1 year ago
 <b>a.c.neumann</b>		0	57		Java	over 3 years ago	about 1 year ago

# Starting from zero Unit Tests



# Separate Test Project





# ZAPs first Unit Test

```
package ch.csnc.extension.util;

import ...

/**
 * Unit test for {@link ch.csnc.extension.util.Encoding}.
 *
 * @author bjoern.kimminich@gmx.de
 */
public class EncodingUnitTest {

    @Test
    public void shouldConvertDataIntoCorrectBase64String() {
        assertThat(Encoding.base64encode("Hello World".getBytes()), is(equalTo("SGVsbG8gV29ybGQ=")));
    }

    @Test
    public void shouldConvertBase64StringIntoCorrectData() {
        assertThat(Encoding.base64decode("SGVsbG8gV29ybGQ="), is(equalTo("Hello World".getBytes())));
    }

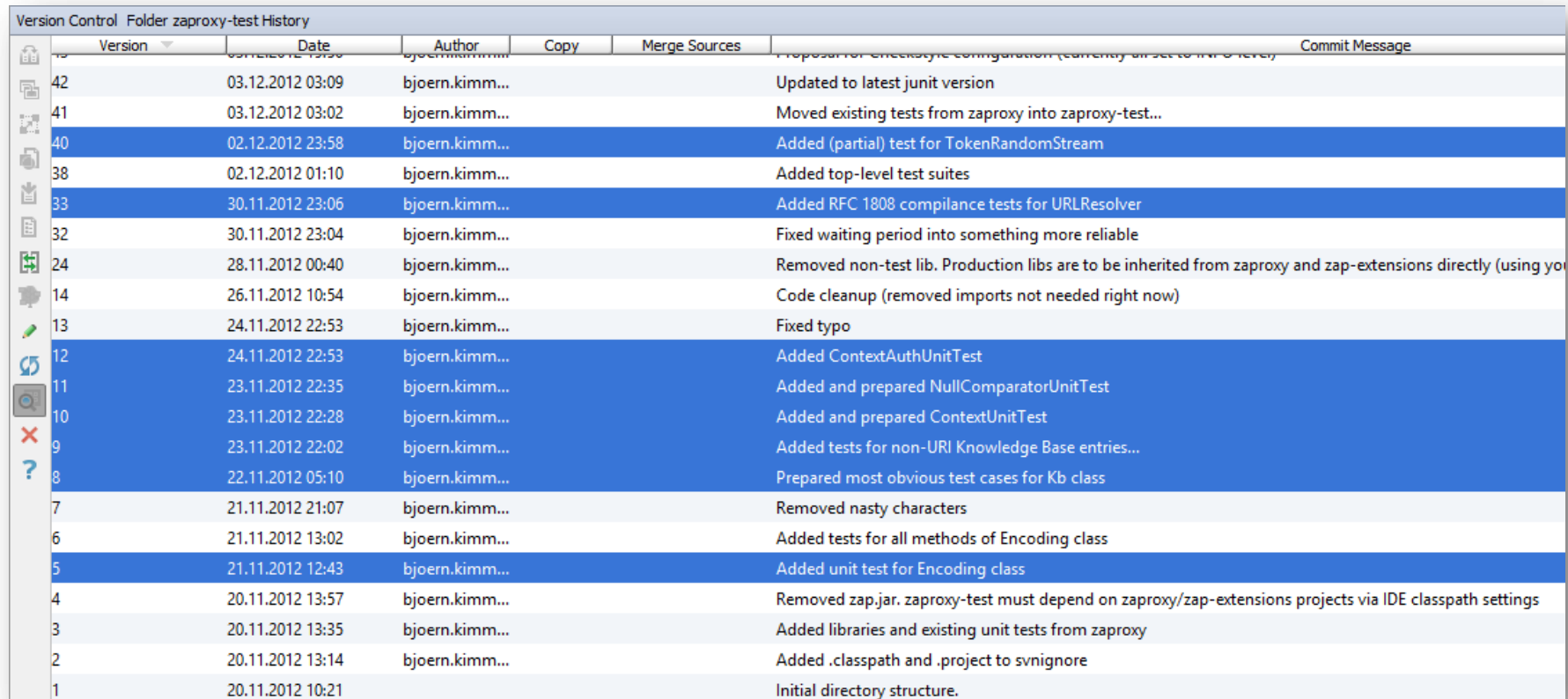
    @Test
    public void shouldConvertDataIntoCorrectHexString() {
        assertThat(Encoding.toHexString("Hello World".getBytes()), is(equalTo("48656c6c6f20576f726c64")));
    }

    @Test
    public void shouldConvertStringIntoCorrectMD5Hash() {...}

    @Test
    public void shouldConvertStringIntoCorrectSHAHash() {...}

    @Test
    public void shouldConvertStringIntoCorrectRot13Cipher() {...}
```

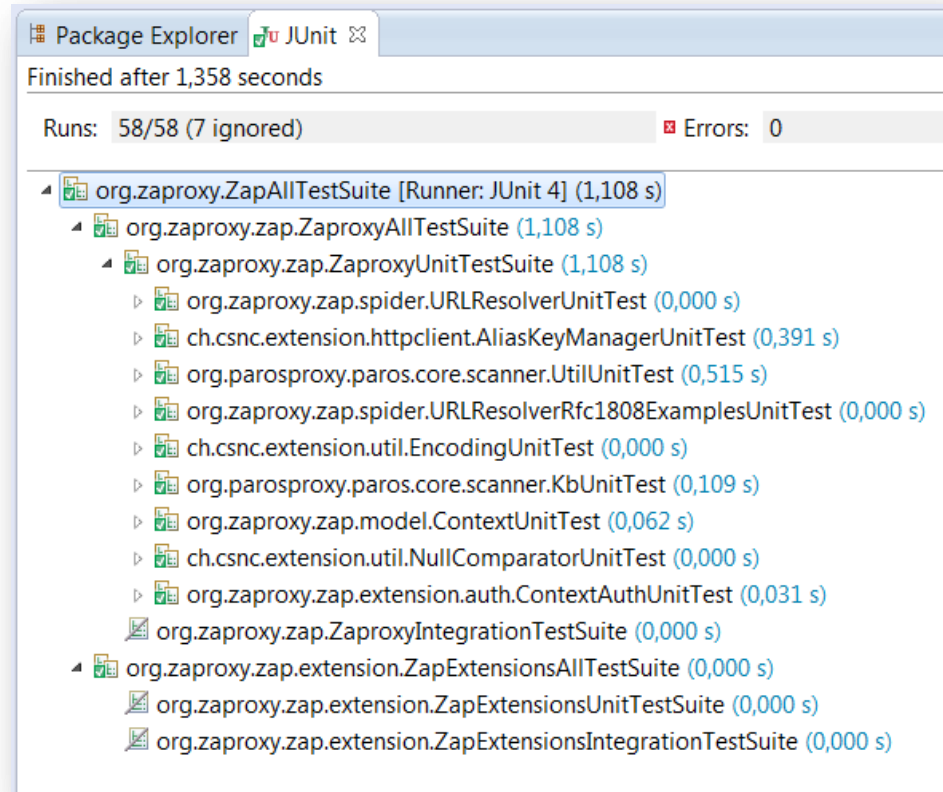
# Adding some more Show Cases



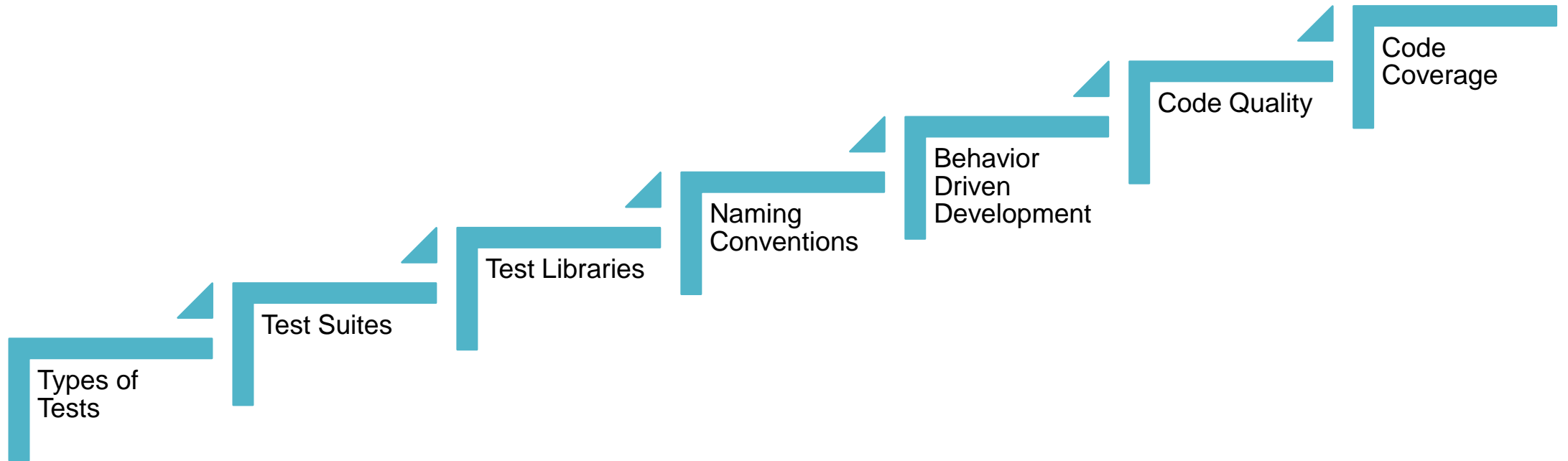
The screenshot shows a version control history window titled "Version Control Folder zapproxy-test History". The window contains a table with columns for Version, Date, Author, Copy, Merge Sources, and Commit Message. The table lists 42 commits, with the most recent at the top. The commits are authored by bjoern.kimm... and include various changes such as updating junit, moving tests, adding test suites, and fixing typos. The table is filtered to show only the "Commit Message" column, and the rows are sorted by date in descending order.

Version	Date	Author	Copy	Merge Sources	Commit Message
42	03.12.2012 03:09	bjoern.kimm...			Updated to latest junit version
41	03.12.2012 03:02	bjoern.kimm...			Moved existing tests from zapproxy into zapproxy-test...
40	02.12.2012 23:58	bjoern.kimm...			Added (partial) test for TokenRandomStream
38	02.12.2012 01:10	bjoern.kimm...			Added top-level test suites
33	30.11.2012 23:06	bjoern.kimm...			Added RFC 1808 compliance tests for URLResolver
32	30.11.2012 23:04	bjoern.kimm...			Fixed waiting period into something more reliable
24	28.11.2012 00:40	bjoern.kimm...			Removed non-test lib. Production libs are to be inherited from zapproxy and zap-extensions directly (using yo
14	26.11.2012 10:54	bjoern.kimm...			Code cleanup (removed imports not needed right now)
13	24.11.2012 22:53	bjoern.kimm...			Fixed typo
12	24.11.2012 22:53	bjoern.kimm...			Added ContextAuthUnitTest
11	23.11.2012 22:35	bjoern.kimm...			Added and prepared NullComparatorUnitTest
10	23.11.2012 22:28	bjoern.kimm...			Added and prepared ContextUnitTest
9	23.11.2012 22:02	bjoern.kimm...			Added tests for non-URI Knowledge Base entries...
8	22.11.2012 05:10	bjoern.kimm...			Prepared most obvious test cases for Kb class
7	21.11.2012 21:07	bjoern.kimm...			Removed nasty characters
6	21.11.2012 13:02	bjoern.kimm...			Added tests for all methods of Encoding class
5	21.11.2012 12:43	bjoern.kimm...			Added unit test for Encoding class
4	20.11.2012 13:57	bjoern.kimm...			Removed zap.jar, zapproxy-test must depend on zapproxy/zap-extensions projects via IDE classpath settings
3	20.11.2012 13:35	bjoern.kimm...			Added libraries and existing unit tests from zapproxy
2	20.11.2012 13:14	bjoern.kimm...			Added .classpath and .project to svnignore
1	20.11.2012 10:21	bjoern.kimm...			Initial directory structure.

# Separation into Test Suites



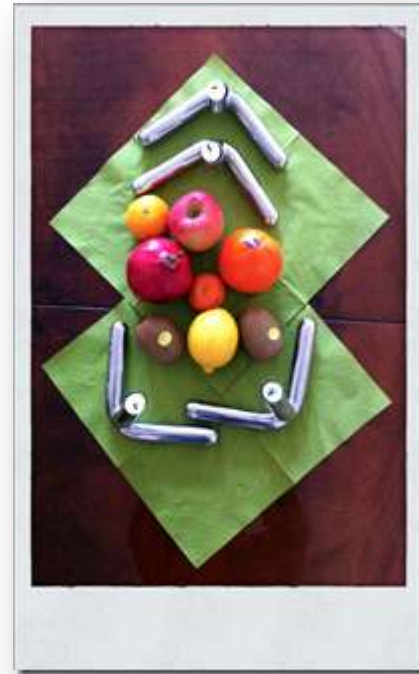
# Providing Test Guidelines



# Pull vs. Push



Pull



Push



# Measure Code Coverage

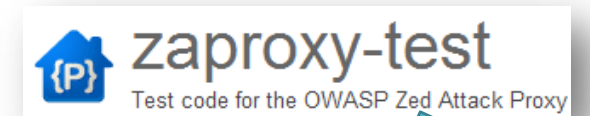
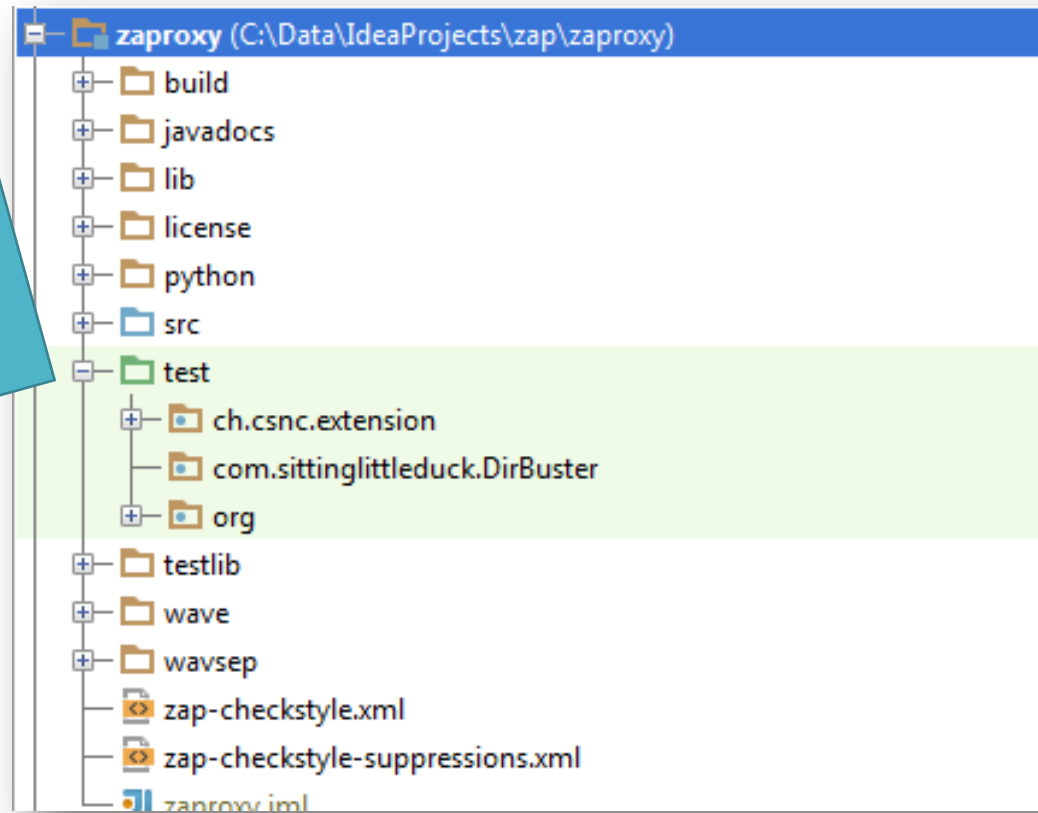
Coverage All in zaproxy (2)

Coverage Summary for 'all classes in scope': 3% classes, 2% lines covered

Element	Class, %	Method, %	Line, %
org.parosproxy.paros	100% (1/1)	10% (3/30)	8% (30/354)
org.zaproxy.zap.extensio...	100% (1/1)	36% (7/19)	37% (36/95)
org.zaproxy.zap.extensio...	100% (1/1)	81% (9/11)	84% (21/25)
org.zaproxy.zap.extensio...	100% (1/1)	53% (8/15)	65% (19/29)
org.zaproxy.zap.users	100% (1/1)	57% (15/26)	69% (71/102)
ch.csnc.extension.util	100% (2/2)	92% (12/13)	80% (125/155)
org.zaproxy.zap.extensio...	100% (2/2)	75% (3/4)	85% (17/20)
org.zaproxy.zap.network	100% (2/2)	54% (6/11)	41% (31/75)
org.zaproxy.zap.model	100% (4/4)	36% (31/86)	31% (120/387)
org.zaproxy.zap.spider	100% (4/4)	88% (24/27)	88% (272/307)
org.zaproxy.zap.utils	100% (4/4)	34% (16/47)	27% (50/184)
ch.csnc.extension.httpcli...	100% (5/5)	53% (30/56)	45% (123/273)
org.zaproxy.zap.control	100% (6/6)	41% (31/75)	52% (188/358)
org.parosproxy.paros.cor...	100% (8/8)	64% (31/48)	71% (201/283)
org.parosproxy.paros.net...	71% (5/7)	36% (60/166)	27% (275/1018)
org.parosproxy.paros.co...	50% (1/2)	3% (1/31)	0% (1/195)
org.zaproxy.zap.extensio...	42% (3/7)	9% (6/66)	10% (34/334)
org.parosproxy.paros.mo...	28% (2/7)	2% (5/224)	2% (29/1402)
org.zaproxy.zap.authenti...	25% (2/8)	13% (7/51)	20% (28/139)
org.zaproxy.zap.extensio...	20% (1/5)	2% (1/46)	0% (1/274)
org.parosproxy.paros.view	16% (1/6)	2% (2/92)	0% (5/521)
org.zaproxy.zap.session	16% (1/6)	2% (1/35)	3% (3/98)
org.zaproxy.zap.view	8% (1/12)	3% (5/138)	2% (16/738)
	0% (0/0)	0% (0/0)	0% (0/0)
ch	0% (0/0)	0% (0/0)	0% (0/0)
ch.csnc	0% (0/0)	0% (0/0)	0% (0/0)
ch.csnc.extension	0% (0/0)	0% (0/0)	0% (0/0)

# Move Tests close to Production Code

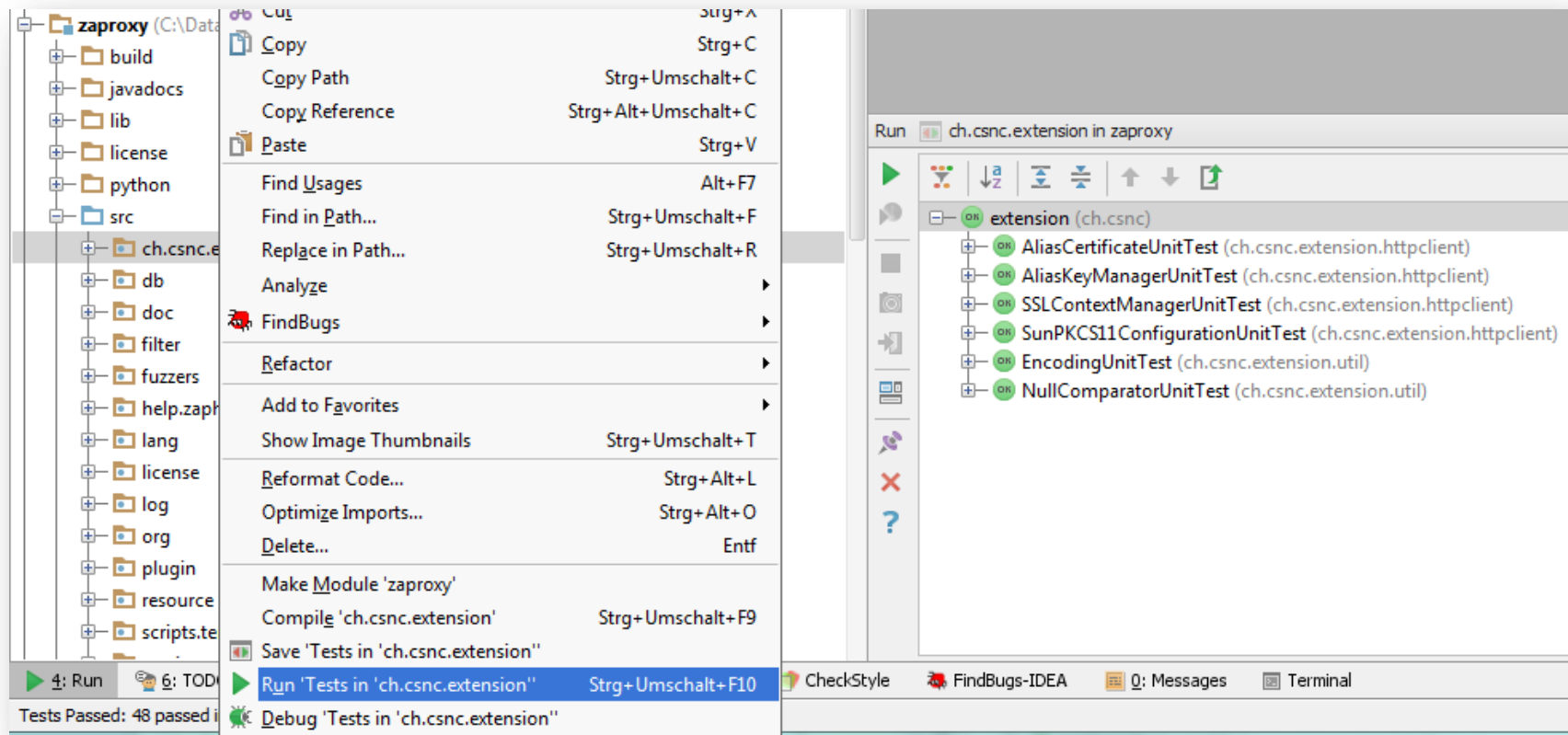
Unit Tests



Build Script

Integration Tests

# Instant execution from IDE

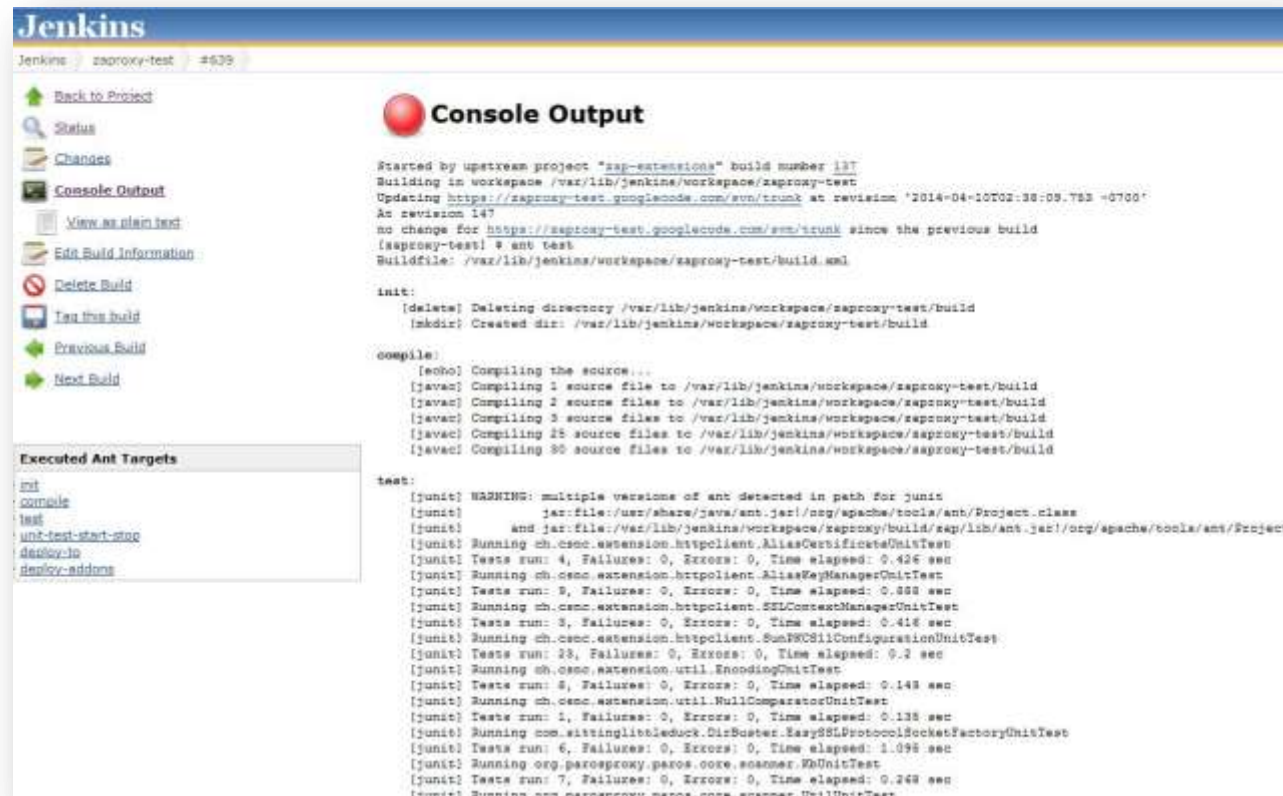




# Run all Tests during Continuous Build...

```
<target name="test" depends="compile">
  <!-- Run the JUnit tests -->
  <junit printsummary="yes" haltonerror="on">
    <classpath>
      <!-- [...] -->
    </classpath>
    <formatter type="plain"/>
    <formatter type="xml"/>
    <batchtest fork="yes" todir="results">
      <fileset dir="${build}">
        <include name="**/*UnitTest.class"/>
        <exclude name="**/Abstract*Test.class"/>
      </fileset>
    </batchtest>
  </junit>
  <!-- [...] -->
</target>
```

...and let it  fail when any tests fail!



The screenshot shows the Jenkins web interface for a build named 'zapproxy-test' (number 4639). The left sidebar contains navigation links: 'Back to Project', 'Status', 'Changes', 'Console Output' (selected), 'View as plain text', 'Edit Build Information', 'Delete Build', 'Tag this build', 'Previous Build', and 'Next Build'. Below these is a section titled 'Executed Ant Targets' with links for 'init', 'compile', 'test', 'unit-test-start-stop', 'deploy-to', and 'deploy-addons'. The main area displays the 'Console Output' for the 'test' target. The output shows the build was triggered by an upstream project, updates the workspace, and then runs the 'test' target. The 'init' target deletes the build directory and creates a new one. The 'compile' target compiles 90 source files. The 'test' target runs a series of JUnit tests, all of which pass with 0 failures and 0 errors. The tests include 'AliasCertificateUnitTest', 'AliasKeyManagerUnitTest', 'SSLContextManagerUnitTest', 'SunPKCS11ConfigurationUnitTest', 'EncodingUnitTest', 'NullComparatorUnitTest', 'EasySSLProtocolSocketFactoryUnitTest', and 'ParosCoreScannerUnitTest'.

```
Started by upstream project "zap-extensions" build number 137
Building in workspace /var/lib/jenkins/workspace/zapproxy-test
Updating https://zapproxy-test.googlecode.com/svn/trunk at revision '2014-04-10T02:38:09.783 -0700'
At revision 147
no change for https://zapproxy-test.googlecode.com/svn/trunk since the previous build
(zapproxy-test) # ant test
Buildfile: /var/lib/jenkins/workspace/zapproxy-test/build.xml

init:
[delete] Deleting directory /var/lib/jenkins/workspace/zapproxy-test/build
[mkdir] Created dir: /var/lib/jenkins/workspace/zapproxy-test/build

compile:
[echo] Compiling the source...
[javac] Compiling 1 source file to /var/lib/jenkins/workspace/zapproxy-test/build
[javac] Compiling 1 source files to /var/lib/jenkins/workspace/zapproxy-test/build
[javac] Compiling 3 source files to /var/lib/jenkins/workspace/zapproxy-test/build
[javac] Compiling 25 source files to /var/lib/jenkins/workspace/zapproxy-test/build
[javac] Compiling 90 source files to /var/lib/jenkins/workspace/zapproxy-test/build

test:
[junit] WARNING: multiple versions of ant detected in path for junit
[junit]   jar file: /usr/share/java/ant.jar/org/apache/tools/ant/Project.class
[junit]   and jar file: /var/lib/jenkins/workspace/zapproxy-test/build/sap/lib/ant.jar/org/apache/tools/ant/Project.
[junit] Running ch.cmc.extension.httpclient.AliasCertificateUnitTest
[junit] Tests run: 4, Failures: 0, Errors: 0, Time elapsed: 0.426 sec
[junit] Running ch.cmc.extension.httpclient.AliasKeyManagerUnitTest
[junit] Tests run: 5, Failures: 0, Errors: 0, Time elapsed: 0.688 sec
[junit] Running ch.cmc.extension.httpclient.SSLContextManagerUnitTest
[junit] Tests run: 3, Failures: 0, Errors: 0, Time elapsed: 0.416 sec
[junit] Running ch.cmc.extension.httpclient.SunPKCS11ConfigurationUnitTest
[junit] Tests run: 13, Failures: 0, Errors: 0, Time elapsed: 0.2 sec
[junit] Running ch.cmc.extension.util.EncodingUnitTest
[junit] Tests run: 5, Failures: 0, Errors: 0, Time elapsed: 0.143 sec
[junit] Running ch.cmc.extension.util.NullComparatorUnitTest
[junit] Tests run: 1, Failures: 0, Errors: 0, Time elapsed: 0.138 sec
[junit] Running com.sittinglittleduck.DirBuster.EasySSLProtocolSocketFactoryUnitTest
[junit] Tests run: 6, Failures: 0, Errors: 0, Time elapsed: 1.098 sec
[junit] Running org.parosproxy.paros.core.scanner.XSSUnitTest
[junit] Tests run: 7, Failures: 0, Errors: 0, Time elapsed: 0.243 sec
[junit] Running org.parosproxy.paros.core.scanner.XSSUnitTest
```

# Future: Adding a GUI Testing Framework

---

ZAP is very UI heavy which makes a lot of the code hard or impossible to unit test

---

Right now there are no GUI Tests in place for ZAP

---

Several free UI Testing Frameworks exist for Java Swing...

---

...unfortunately none is actively maintained any more

**Testing is a crucial** part of Software Development

Good **Tests are** the better **documentation**

**Tests** can **make a difference**  
between a prospering and a dead-end  
OSS project

Conclusion

# Thank you!

**Björn Kimminich**

Web: <http://kimminich.de>

Twitter: [@bkimminich](https://twitter.com/bkimminich)



Background Image: [Eikira](#)