Quo pertentas, OSS?

How Open Source can benefit from well-crafted Tests

Björn Kimminich

Web: http://kimminich.de

Twitter: @bkimminich

v1.0

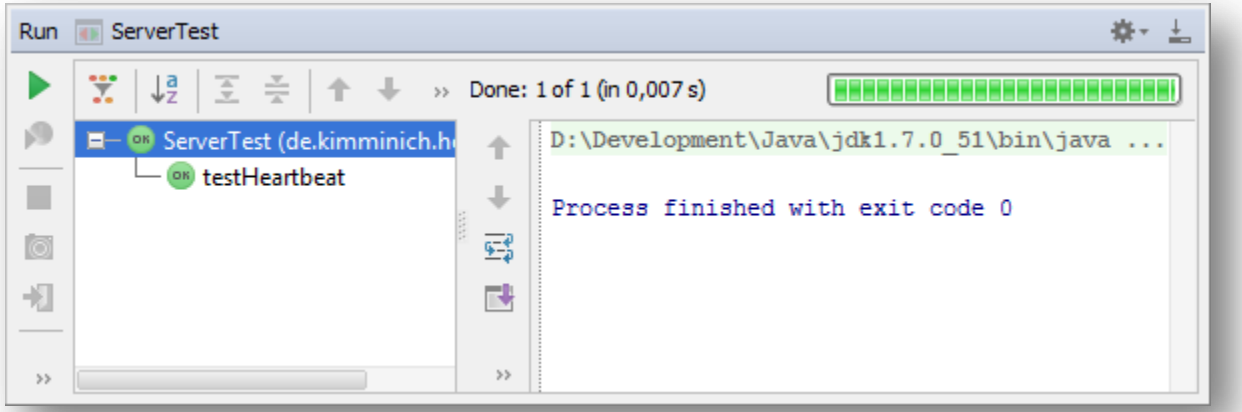
Let‘s start with some code…

```java
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…



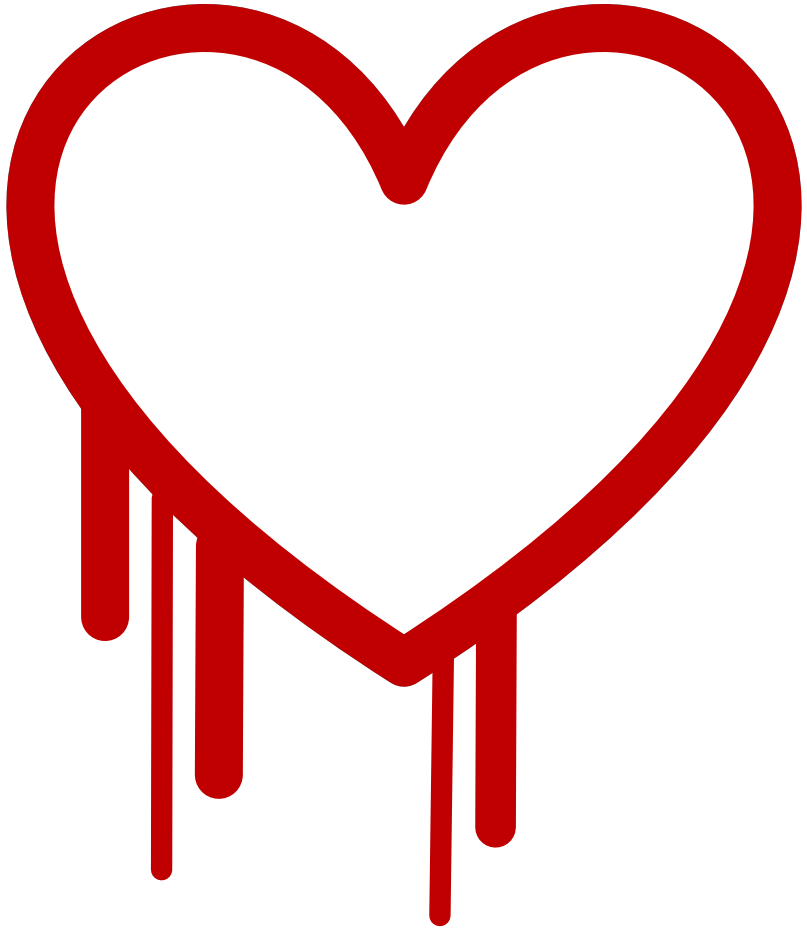
… and achieves 100% code coverage!

Coverage ServerTest

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!

Run ServerTest

Done: 2 of 2 Failed: 1 (in 0,011 s)

ServerTest (de.kimminich.heartbleed)
- testHeartbeat
- testHeartbleed

```
D:\Development\Java\jdk1.7.0_51\bin\java ...

org.junit.ComparisonFailure:
Expected :1
Actual   :1111608098;user=bjoern.kimminich&password=s3cret123!;de.kimminic
   <Click to see difference>

<2 internal calls>
    at de.kimminich.heartbleed.ServerTest.testHeartbleed(ServerTest.java:23) <23 internal calls>


Process finished with exit code -1
```

Finding Bugs in Open Source Software

Code Reviews

Pair Programming

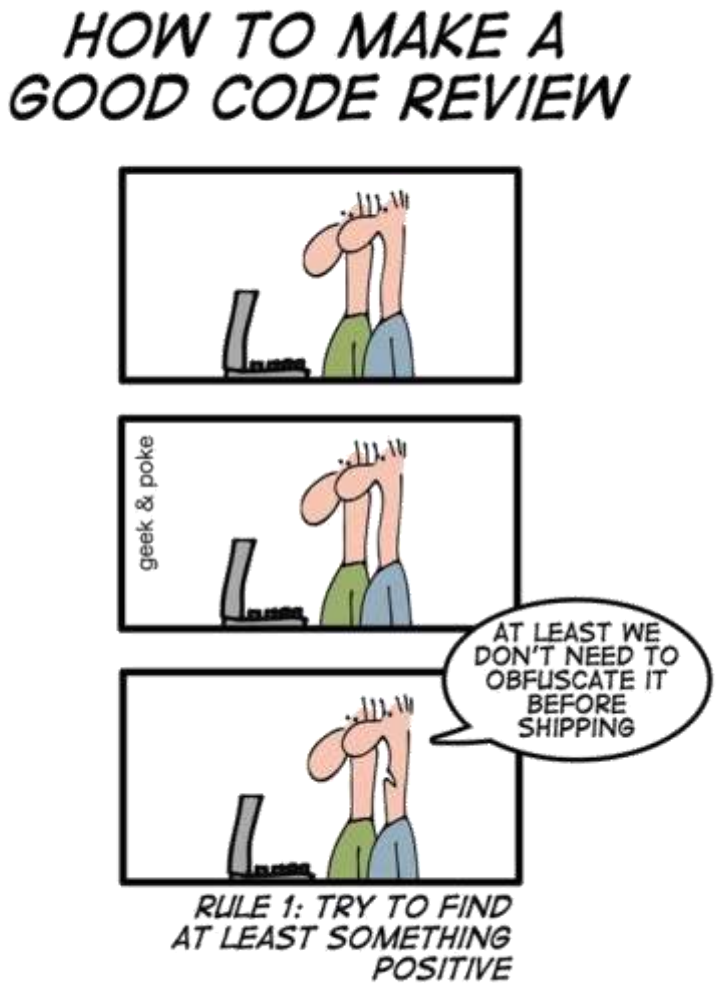
- Infeasible with remote development
- Occasionally during Hackathons

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



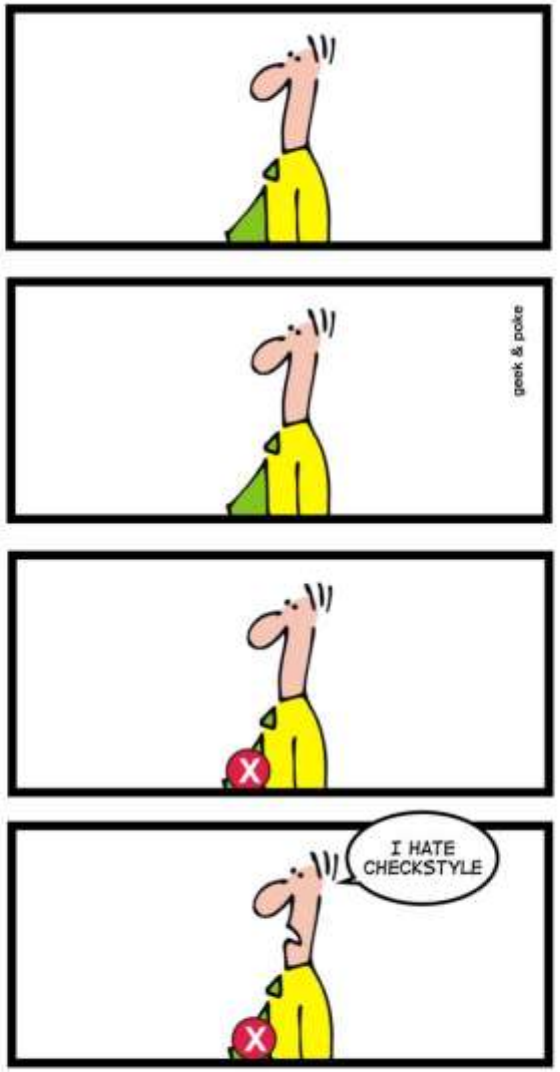
Cartoon: Geek & Poke

Static Code Analysis

Popular Open Source Tools			
FindBugs	CheckStyle	PMD	Sonar

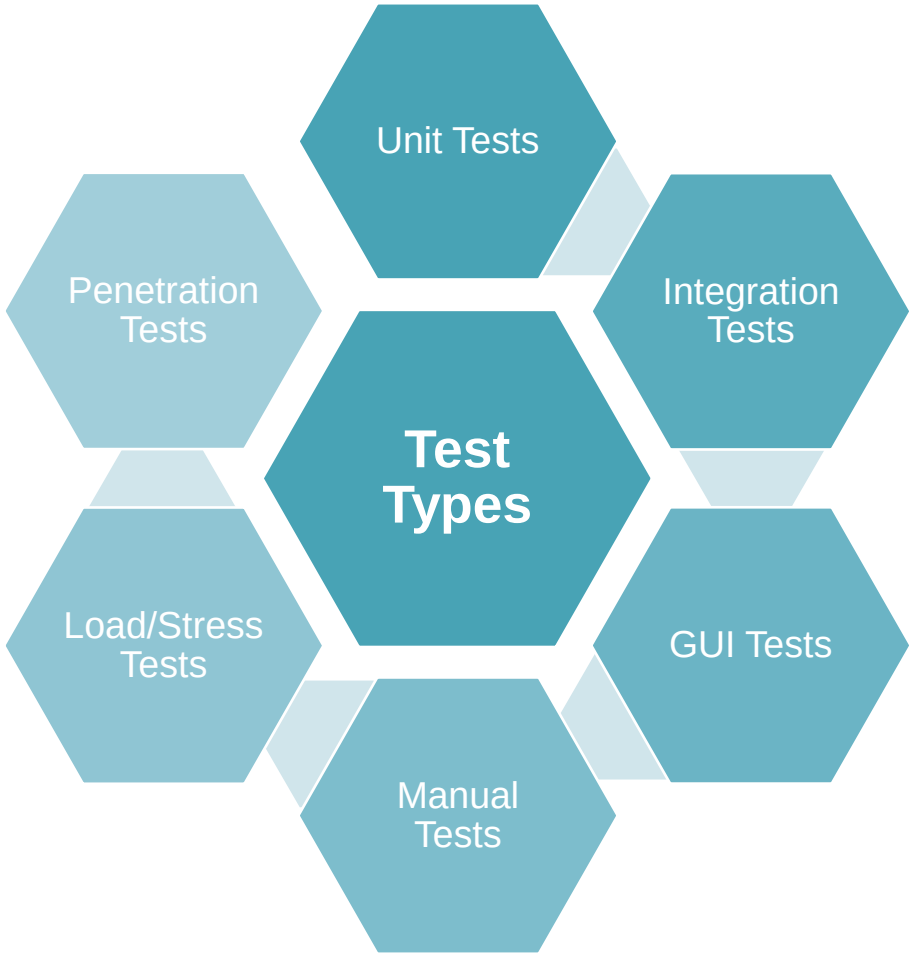
Find code smells and potential programming errors…	
…but miss a lot as well	…or produce false positives

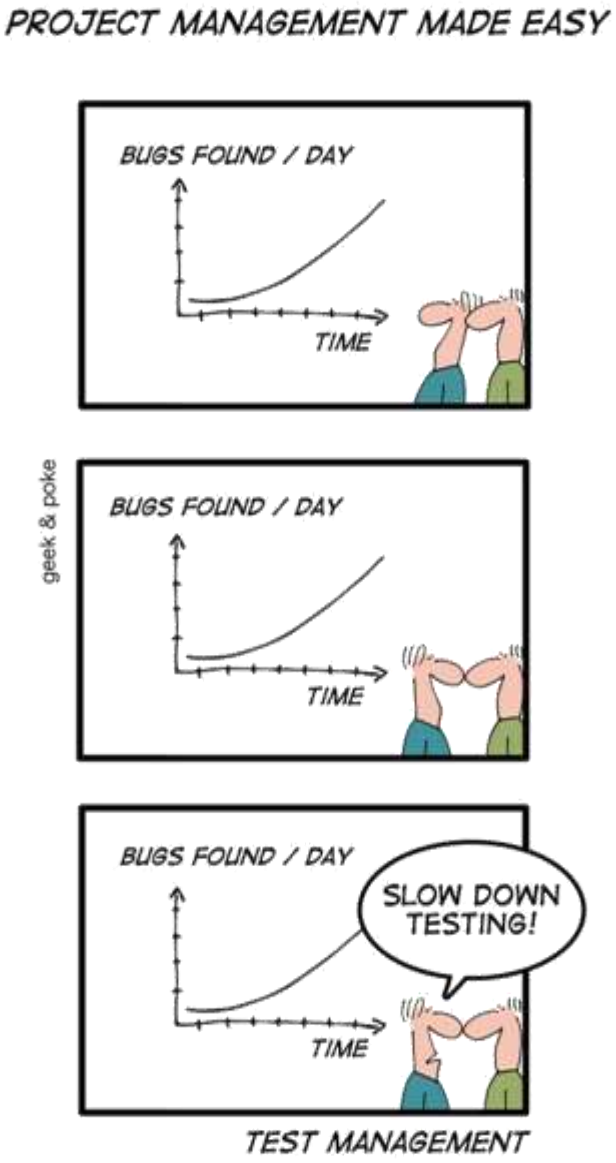
Some commercial Tools might be more powerful…
…but are typically not affordable for OSS projects



Cartoon: Geek & Poke

Testing



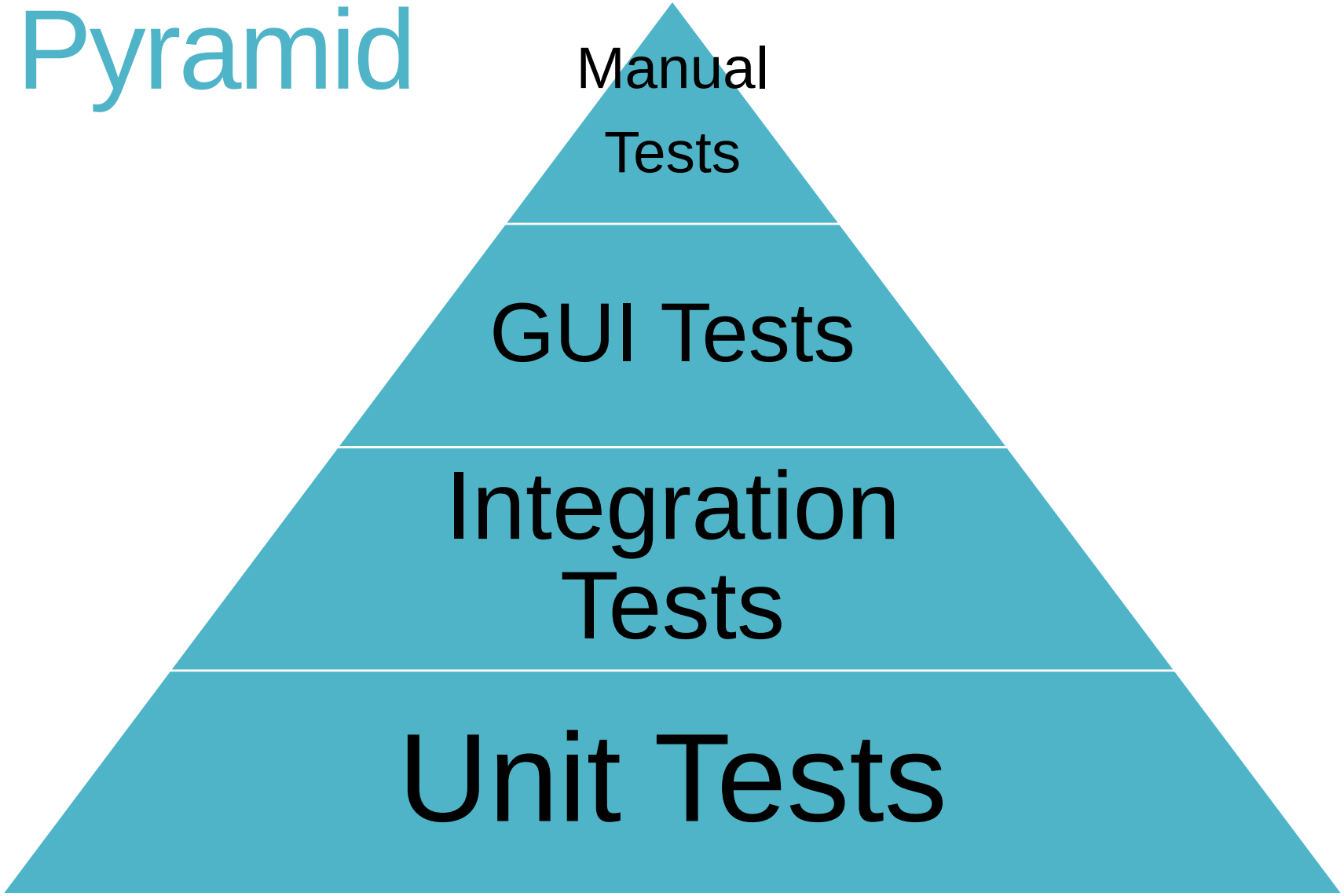


Cartoon: [Geek & Poke]

Best vs. Bad Practices for Testing

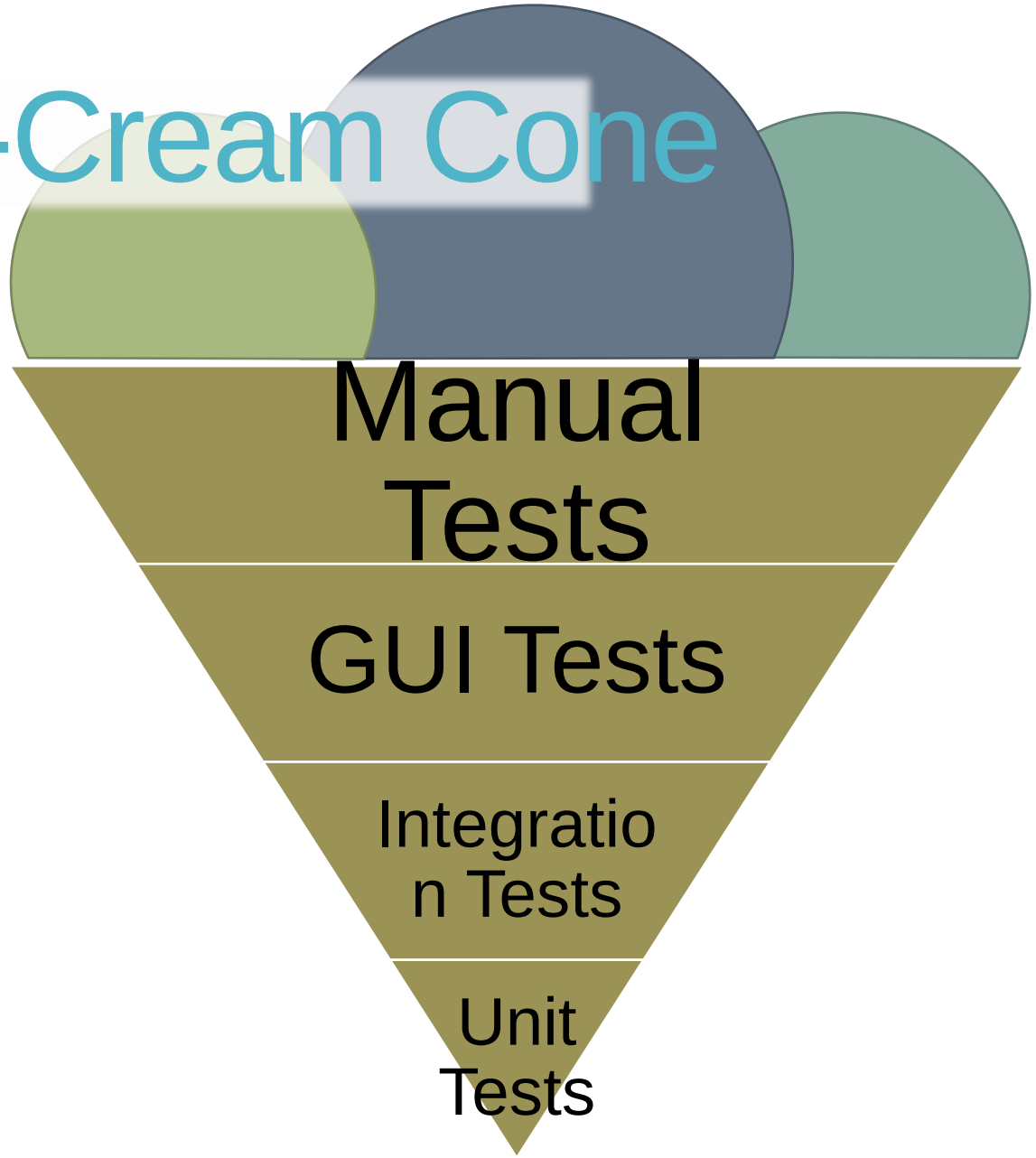
Test Pyramid

BEST PRACTICE



Source: WatirMelon

Test Ice-Cream Cone



BAD PRACTICE

Source: WatirMelon

Happy Path Testing

BAD PRACTICE

Photo: Tortured Mind Photography

Testing Border & Exceptional Cases

BEST 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

BAD PRACTICE

```
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

BAD PRACTICE

Scenario Tests with BDD

BEST 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()));
}
```

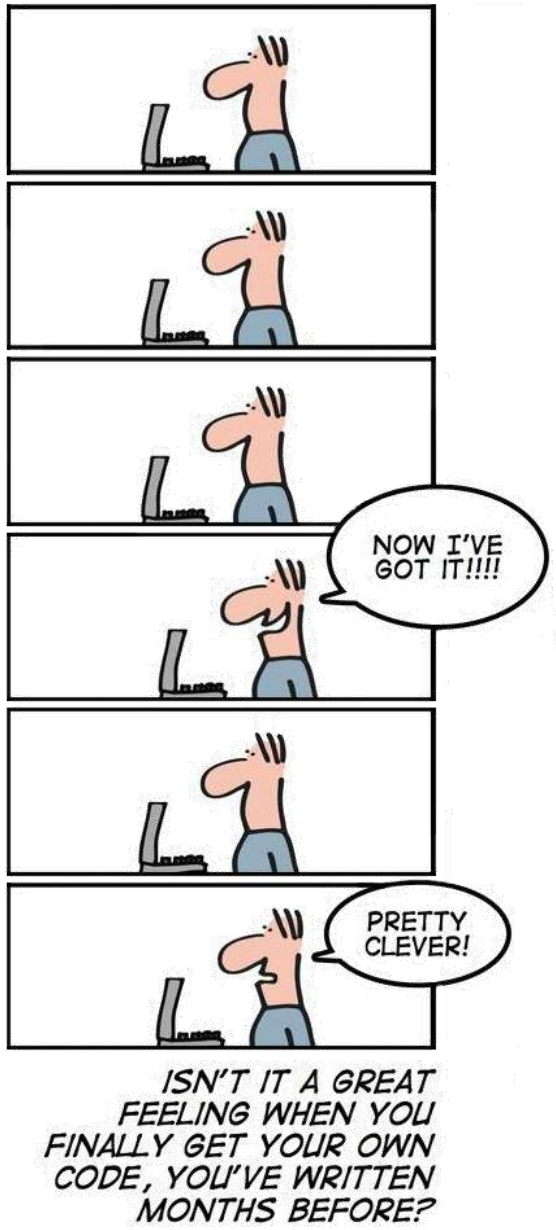
Benefits of well-crafted Tests for OSS

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!



Cartoon: Geek & Poke

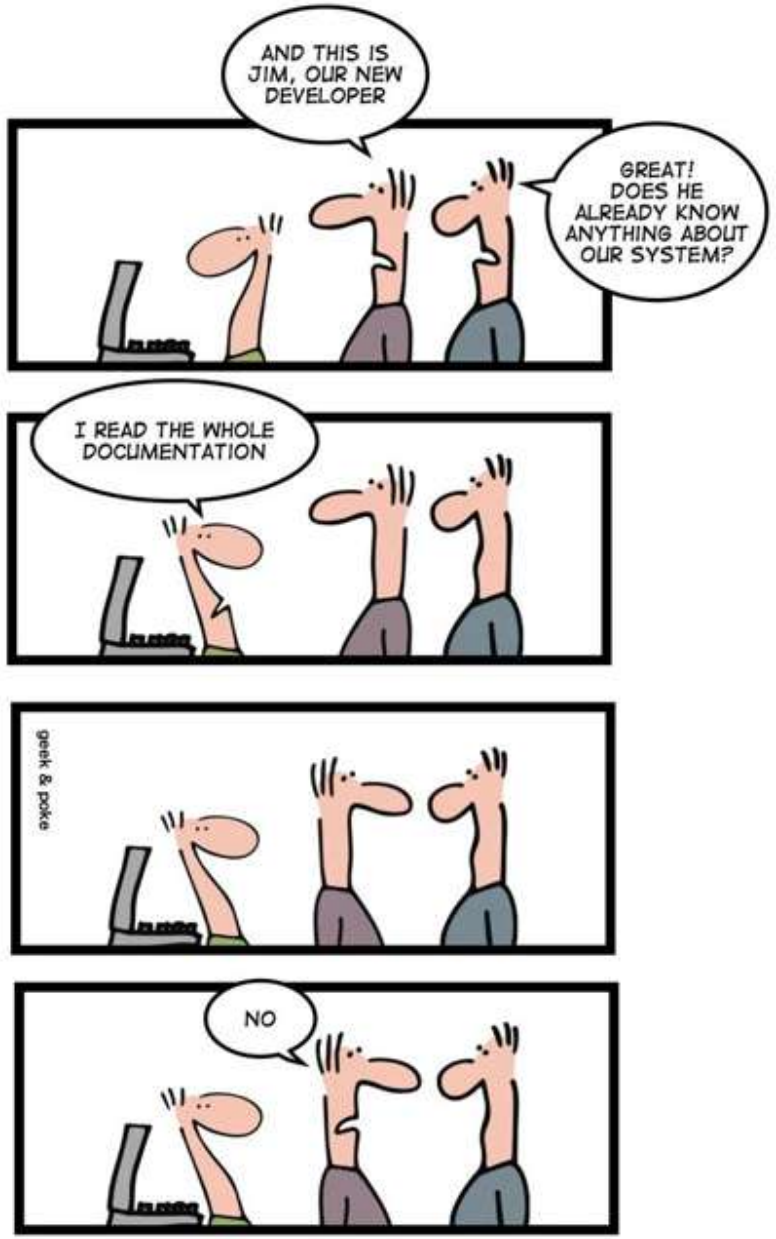
Documentation++

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



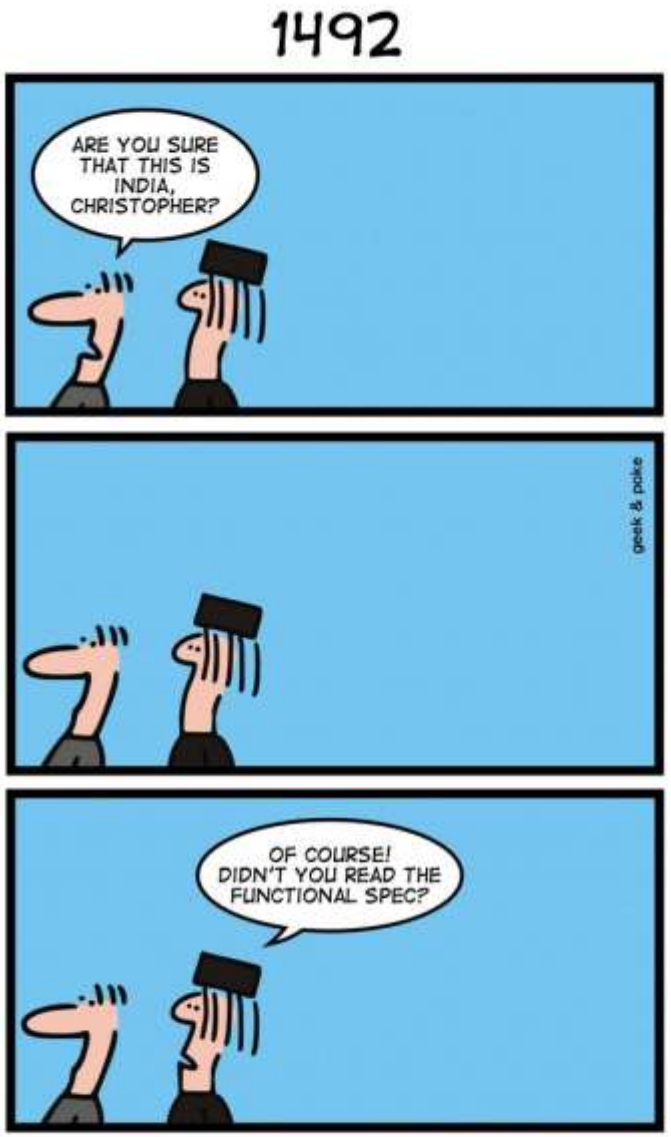
Cartoon: Geek & Poke

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)

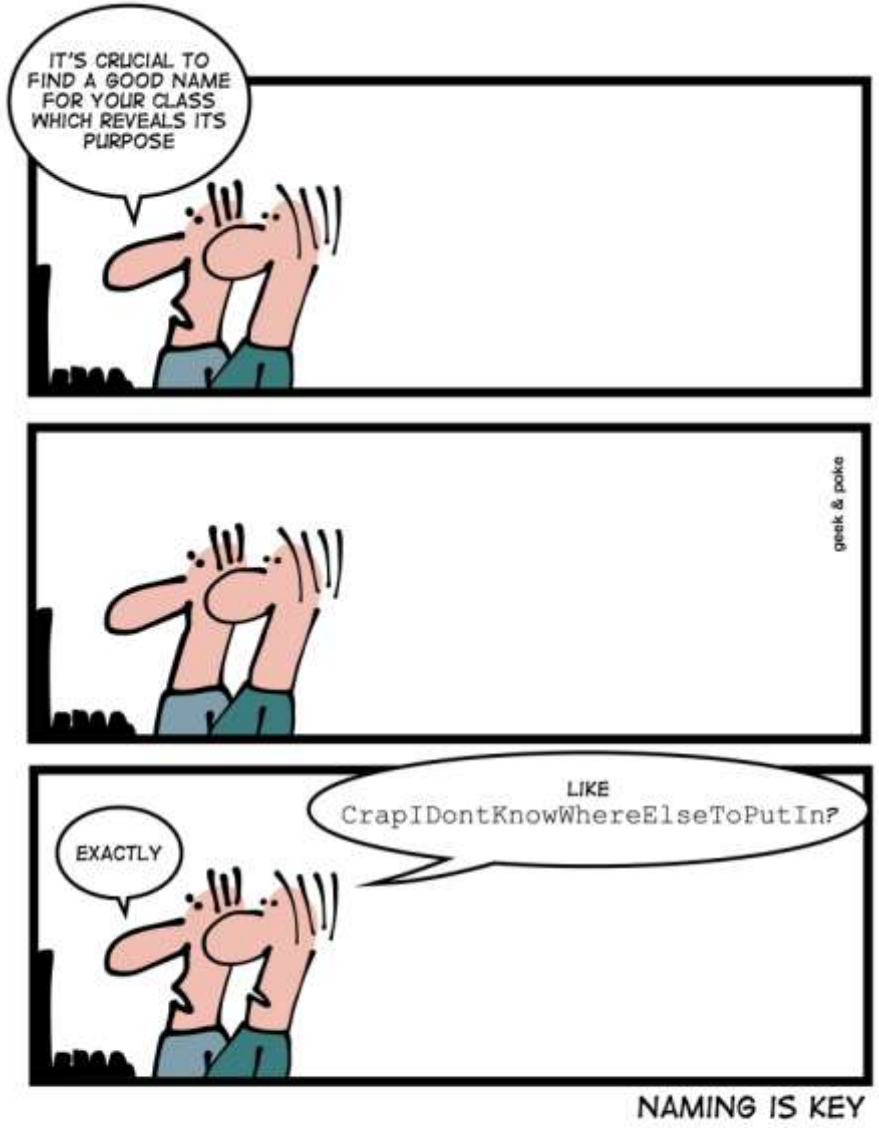


Cartoon: Geek & Poke

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)



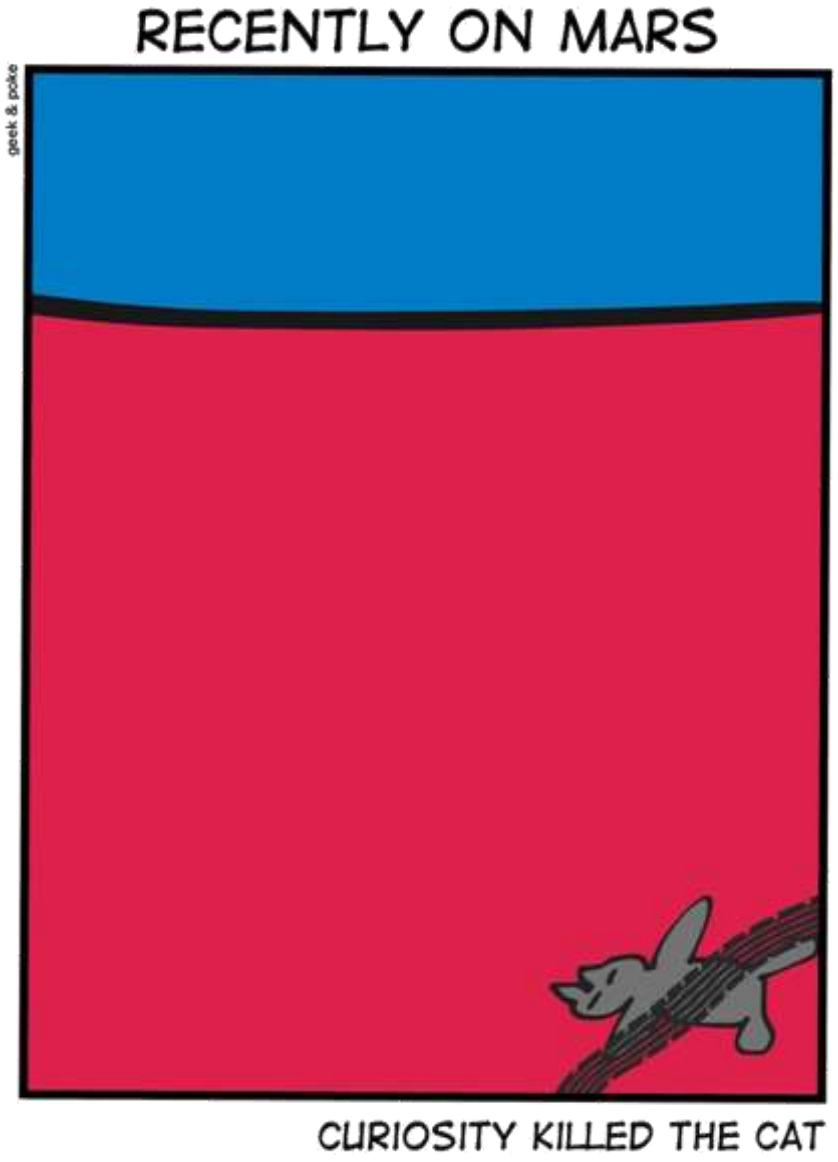
Cartoon: Geek & Poke

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



Cartoon: Geek & Poke

Introducing Unit Tests to OWASP ZAP

OWASP Zed Attack Proxy (ZA

Easy-to-use integrated penetration-testing tool

- Locates vulnerabilities in web applications
- Helps building secure apps
- OWASP Flagship Project
- Programmed in Java with *javax.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

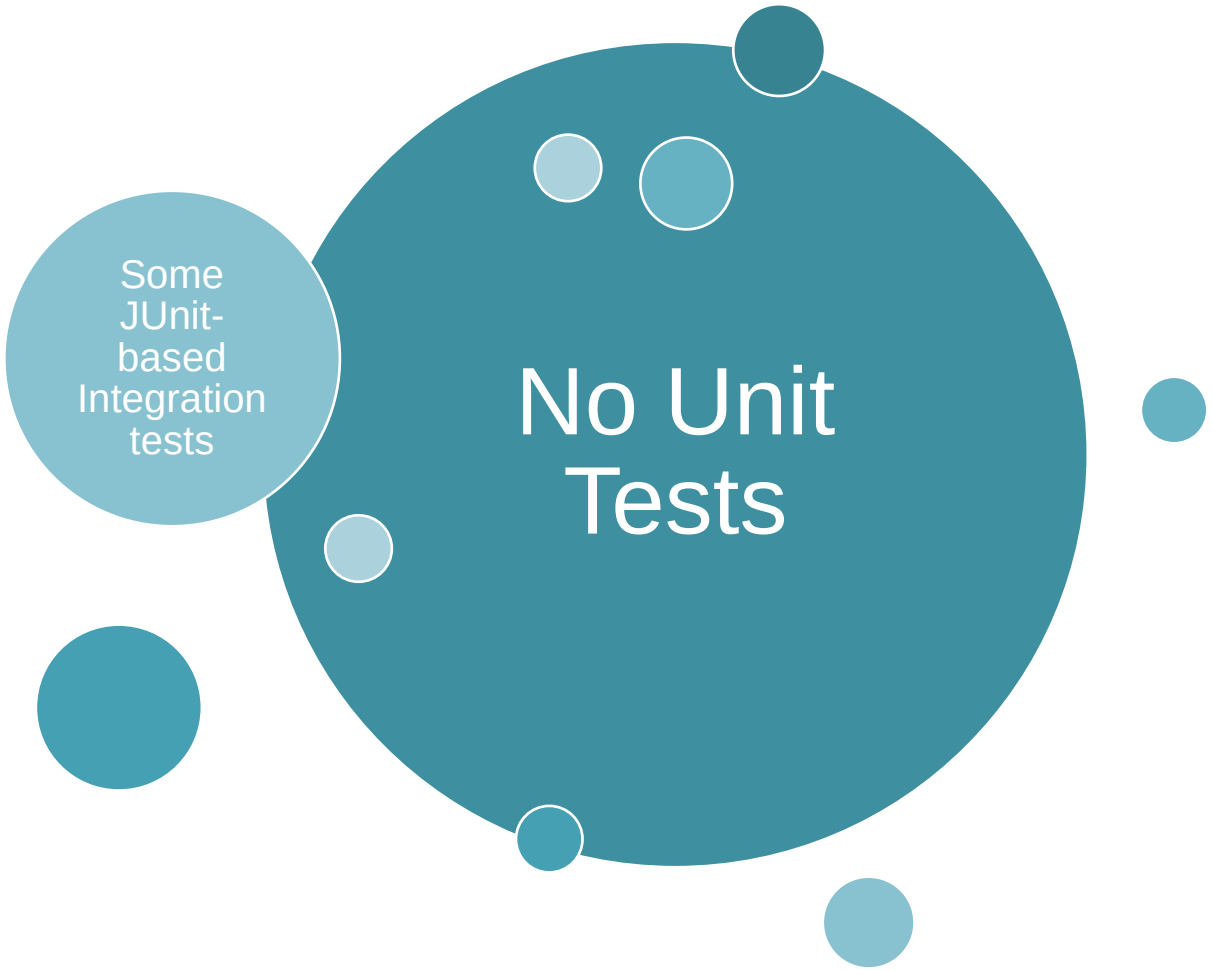
Promote ZAP
https://code.google.com/p/zaproxy/wiki/ZapEvangelists

ZAP Truck Factor ≤2

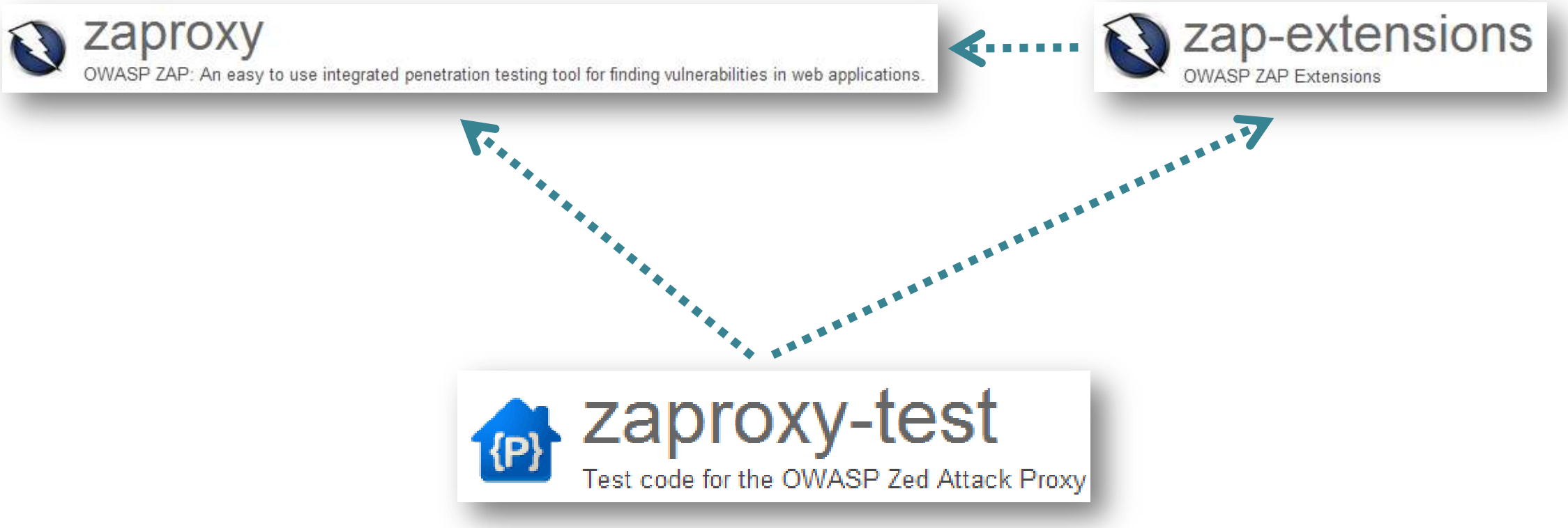
Name	Kudos	12 Month Commits	All Time Commits	5 Year Trend	Primary Language	First Commit	Last Commit
Simon Bennetts (ZAP Project Lead)	9	693	1607		Java	over 3 years ago	6 days ago
THC...@gmail.com	8	438	719		Java	over 2 years ago	6 days ago
colm.p.o...@gmail.com	8	29	241		Java	almost 2 years ago	8 days ago
Cosmin Stefan Dobrin	8	127	186		Java	almost 2 years ago	25 days ago
yha...@gmail.com	8	176	176		Java	10 months ago	26 days ago
Björn Kimminich	8	40	97		Java	over 1 year ago	30 days ago
maw...@ymail.com	7	0	76		Java	about 3 years ago	over 1 year ago
70poi...@gmail.com	7	74	74		Java	4 months ago	8 days ago
desousa.vitor	7	0	67		Java	about 2 years ago	about 2 years ago
leandro...@talsoft.c...	7	0	59		Java	over 1 year ago	over 1 year ago
a.c.neumann	7	0	57		Java	over 3 years ago	about 1 year ago

Source: Ohloh

Starting from zero Unit Tests



Separate Test Project



ZAPs first Unit Test

```java
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

Version Control Folder zaproxy-test History

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 zaproxy into zaproxy-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 compilance 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 zaproxy 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. zaproxy-test must depend on zaproxy/zap-extensions projects via IDE classpath settings
3	20.11.2012 13:35	bjoern.kimm...			Added libraries and existing unit tests from zaproxy
2	20.11.2012 13:14	bjoern.kimm...			Added .classpath and .project to svnignore
1	20.11.2012 10:21				Initial directory structure.

Separation into Test Suites

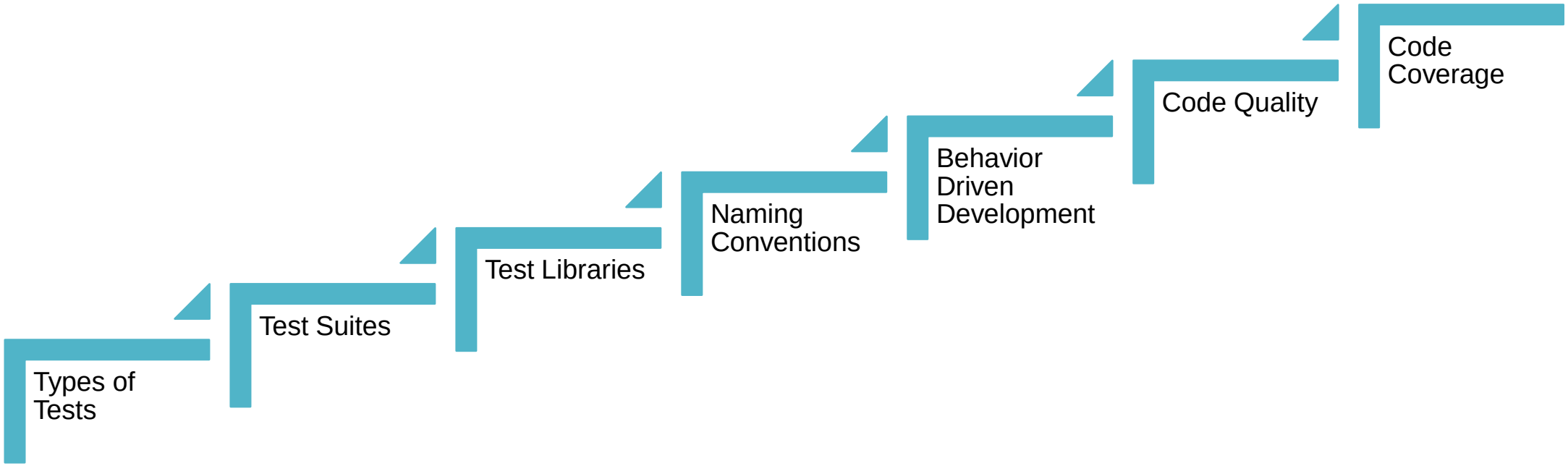
Package Explorer | JUnit

Finished after 1,358 seconds

Runs: 58/58 (7 ignored) Errors: 0

- org.zaproxy.ZapAllTestSuite [Runner: JUnit 4] (1,108 s)
  - org.zaproxy.zap.ZaproxyAllTestSuite (1,108 s)
    - org.zaproxy.zap.ZaproxyUnitTestSuite (1,108 s)
      - org.zaproxy.zap.spider.URLResolverUnitTest (0,000 s)
      - ch.csnc.extension.httpclient.AliasKeyManagerUnitTest (0,391 s)
      - org.parosproxy.paros.core.scanner.UtilUnitTest (0,515 s)
      - org.zaproxy.zap.spider.URLResolverRfc1808ExamplesUnitTest (0,000 s)
      - ch.csnc.extension.util.EncodingUnitTest (0,000 s)
      - org.parosproxy.paros.core.scanner.KbUnitTest (0,109 s)
      - org.zaproxy.zap.model.ContextUnitTest (0,062 s)
      - ch.csnc.extension.util.NullComparatorUnitTest (0,000 s)
      - org.zaproxy.zap.extension.auth.ContextAuthUnitTest (0,031 s)
    - org.zaproxy.zap.ZaproxyIntegrationTestSuite (0,000 s)
  - org.zaproxy.zap.extension.ZapExtensionsAllTestSuite (0,000 s)
    - org.zaproxy.zap.extension.ZapExtensionsUnitTestSuite (0,000 s)
    - org.zaproxy.zap.extension.ZapExtensionsIntegrationTestSuite (0,000 s)

Providing Test Guidelines



Pull vs. Push

Pull

Push

Photos: [One Man Think Tank]

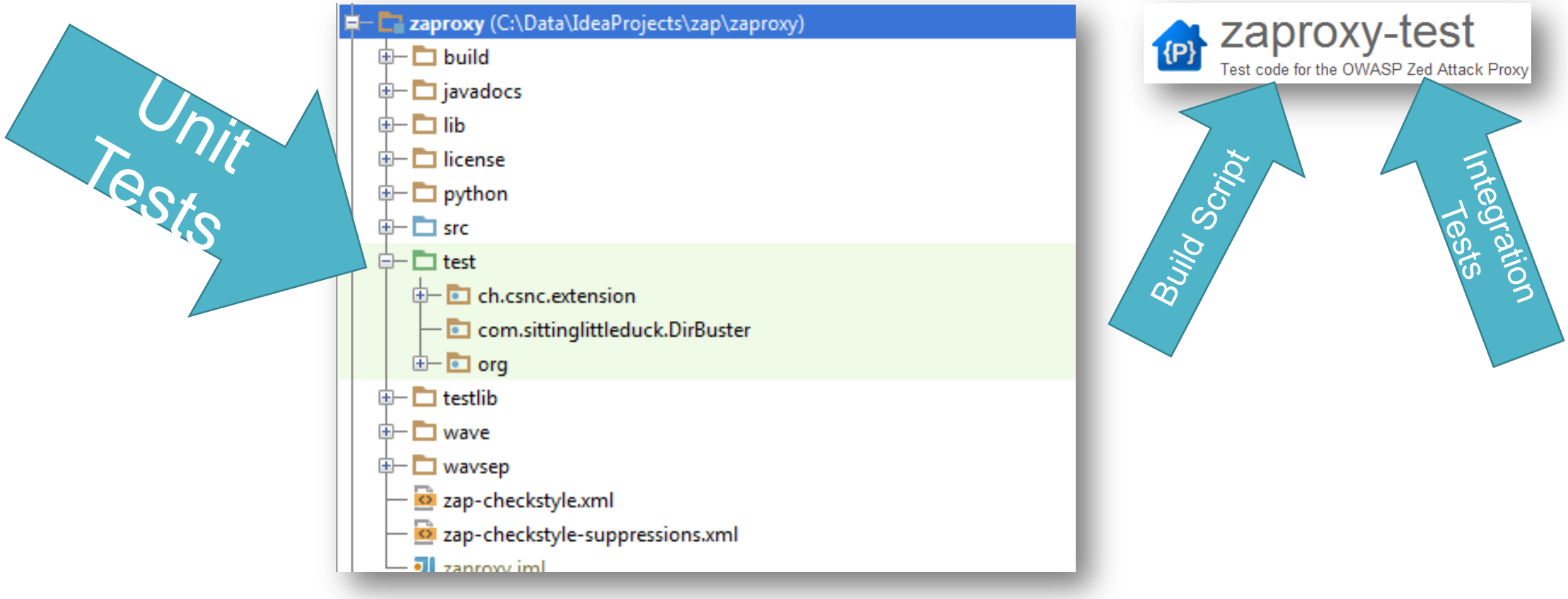
Measure Code Coverage

Coverage All in zaproxy (2)

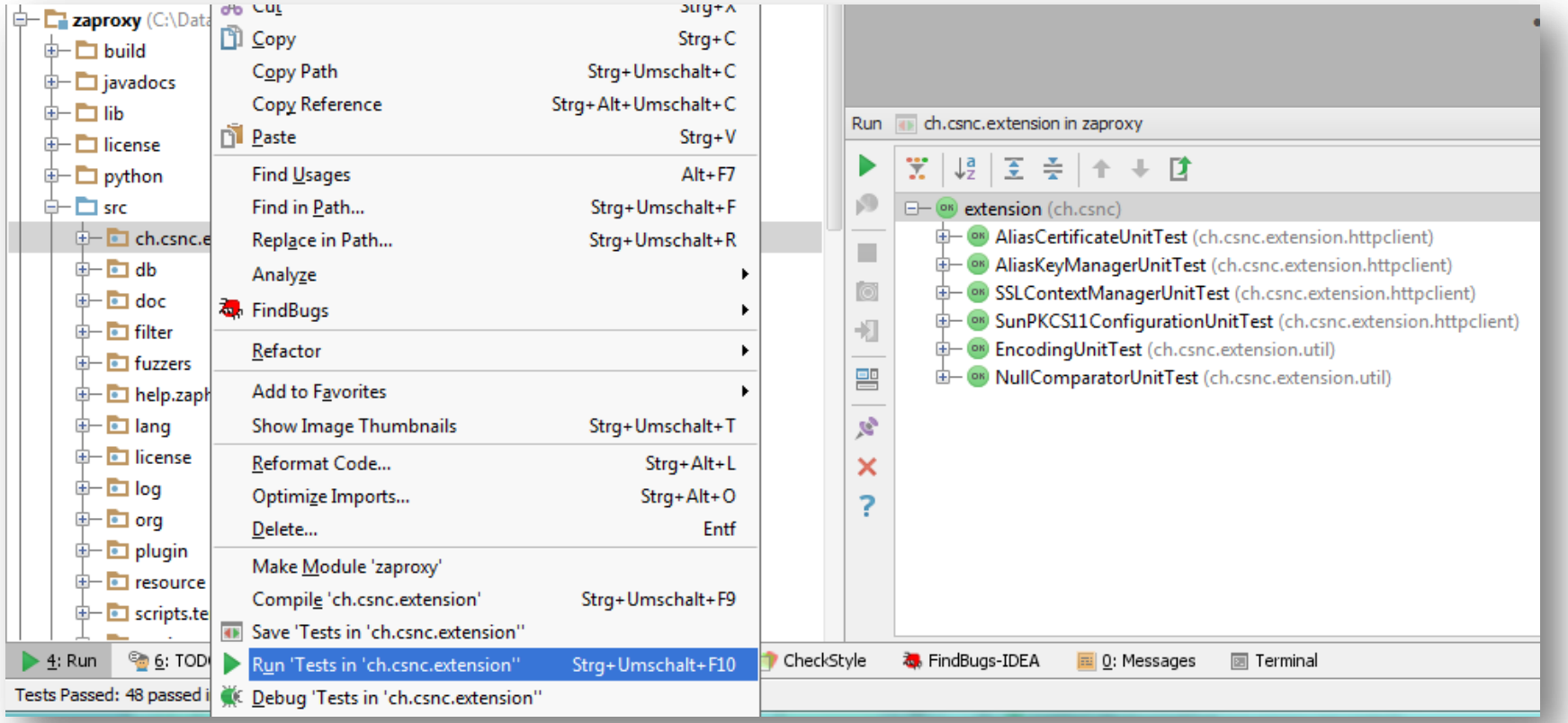
Coverage Summary for 'all classes in scope': 3% classes, 2% lines covered

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)

Move Tests close to Production Code



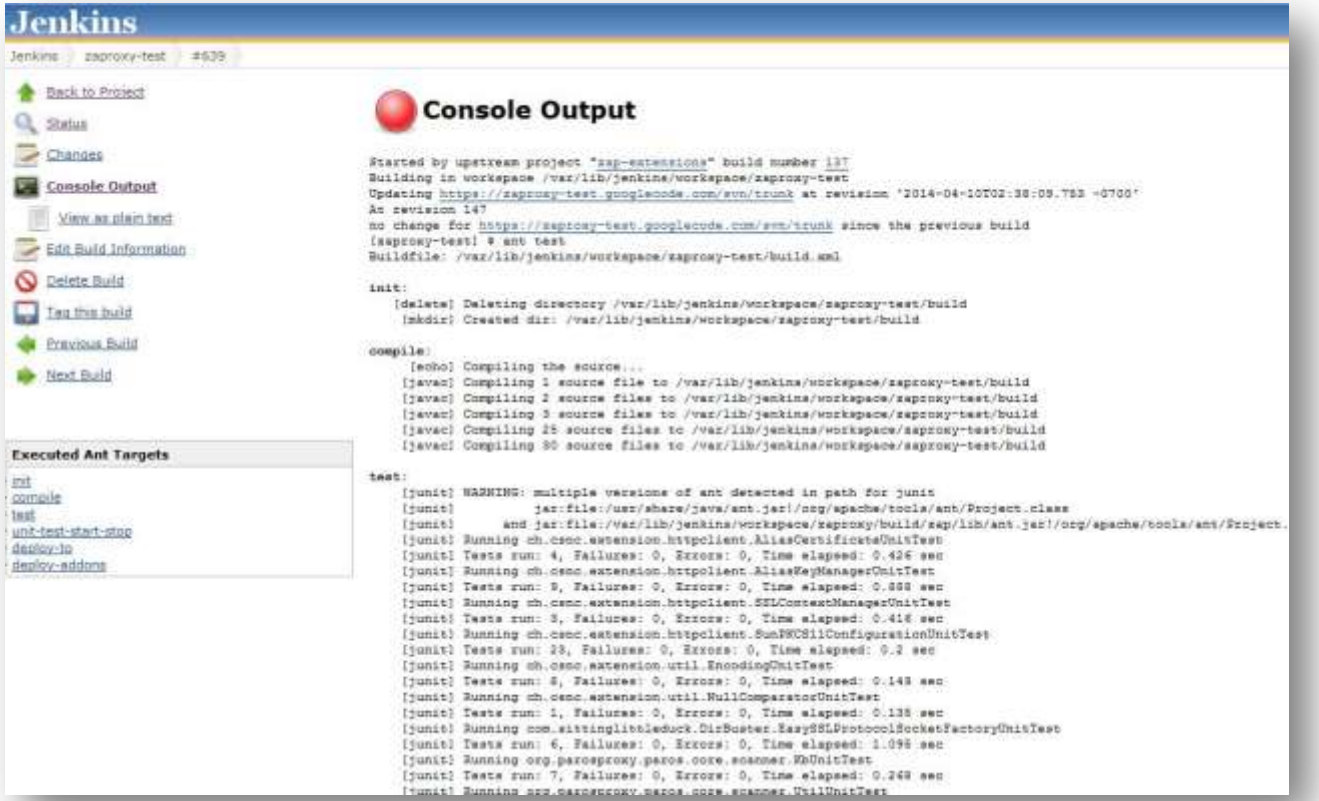
Instant execution from IDE



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!



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

Conclusion

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

Thank you!

Björn Kimminich
Web: http://kimminich.de
Twitter: @bkimminich

Background Image: Eikira